

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re PATENT APPLICATION of - :

Takayuki KONDO :

Serial No.: [NEW] : Attn: Applications Branch

Filed: July 18, 2003 : Attorney Docket No.: OKI.551

For: FAILURE ANALYSIS SYSTEM AND FAILURE ANALYSIS METHOD OF LOGIC  
LSI

**CLAIM OF PRIORITY**

Honorable Assistant Commissioner for Patents and Trademarks,  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

Applicant, in the above-identified application, hereby claims the priority date  
under the International Convention of the following Japanese application:

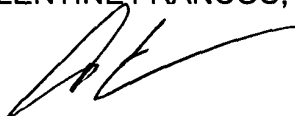
Appln. No. 2002-259280 filed September 4, 2002

as acknowledged in the Declaration of the subject application.

A certified copy of said application is being submitted herewith.

Respectfully submitted,

VOLENTINE FRANCOS, PLLC



Adam C. Volentine  
Registration No. 33,289

12200 Sunrise Valley Drive, Suite 150  
Reston, Virginia 20191  
Tel. (703) 715-0870  
Fax. (703) 715-0877

Date: July 18, 2003

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 9月 4日

出 願 番 号

Application Number:

特願2002-259280

[ ST.10/C ]:

[ JP2002-259280 ]

出 願 人

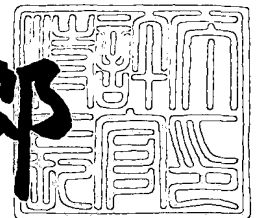
Applicant(s):

沖電気工業株式会社  
株式会社 沖マイクロデザイン

2003年 2月14日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号 出証特2003-3006952

【書類名】 特許願

【整理番号】 KT000419

【提出日】 平成14年 9月 4日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 G06F 11/00

【発明者】

    【住所又は居所】 宮崎県宮崎郡清武町大字木原 7 0 8 3 番地 株式会社沖  
                                マイクロデザイン内

    【氏名】 近藤 高行

【特許出願人】

    【識別番号】 000000295

    【氏名又は名称】 沖電気工業株式会社

【特許出願人】

    【識別番号】 591049893

    【氏名又は名称】 株式会社沖マイクロデザイン

【代理人】

    【識別番号】 100095957

    【弁理士】

    【氏名又は名称】 亀谷 美明

    【電話番号】 03-5919-3808

【選任した代理人】

    【識別番号】 100096389

    【弁理士】

    【氏名又は名称】 金本 哲男

    【電話番号】 03-3226-6631

【選任した代理人】

    【識別番号】 100101557

    【弁理士】

    【氏名又は名称】 萩原 康司

【電話番号】 03-3226-6631

【手数料の表示】

【予納台帳番号】 040224

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9707549

【包括委任状番号】 9707550

【包括委任状番号】 9707551

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ロジック L S I の不良解析システム及び不良解析方法

【特許請求の範囲】

【請求項 1】 ソフトウェアを内蔵したロジック L S I の不良解析システムであって、

クロックに同期して前記ロジック L S I の端子信号情報を記録する機能と、

前記記録した端子信号情報をクロックに同期して再生する機能と、

前記再生した端子信号情報と、正常なロジック L S I の端子信号情報とを比較する機能と、

を有することを特徴とするロジック L S I の不良解析システム。

【請求項 2】 ソフトウェアを内蔵したロジック L S I の不良解析システムであって、

クロックに同期して解析対象のロジック L S I の端子信号情報を記録する機能と、

クロックに同期して正常なロジック L S I の端子信号情報を記録する機能と、

前記記録した解析対象のロジック L S I の端子信号情報と、前記記録した正常なロジック L S I の端子信号情報とを比較する機能と、

を有することを特徴とするロジック L S I の不良解析システム。

【請求項 3】 ソフトウェアを内蔵したロジック L S I の不良解析システムであって、

クロックに同期して解析対象のロジック L S I の端子信号情報を記録する機能と、

前記記録した端子信号情報をクロックに同期して再生する機能と、

前記再生した解析対象のロジック L S I の端子信号情報と、ロジック L S I のエミュレータの端子信号情報とを比較する機能と、

を有することを特徴とするロジック L S I の不良解析システム。

【請求項 4】 ソフトウェアを内蔵したロジック L S I の不良解析システムであって、

クロックに同期して解析対象のロジック L S I の端子信号情報を記録する機能

と、

前記記録した解析対象のロジック L S I の端子信号情報と、ロジック L S I のエミュレータの端子信号情報とを比較する機能と、  
を有することを特徴とするロジック L S I の不良解析システム。

【請求項 5】 前記端子信号情報は、特定期間におけるレジスタデータと R A M データの状態変化のトレースデータマップを含むことを特徴とする請求項 1 , 2 , 3 , または 4 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 6】 解析対象のロジック L S I のトレースデータマップと、正常なロジック L S I のトレースデータマップとのトレース差分マップを生成する機能をさらに有することを特徴とする請求項 1 , 2 , 3 , 4 , または 5 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 7】 複数の不具合を発生させて、前記トレース差分マップを複数生成し、差分の平均とばらつきを統計処理して求める機能をさらに有することを特徴とする請求項 1 , 2 , 3 , 4 , 5 , または 6 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 8】 不具合発生前の R A M 及びレジスタの状態を保存し、不具合発生前から復活させる機能をさらに有することを特徴とする請求項 1 , 2 , 3 , 4 , 5 , 6 , または 7 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 9】 C P U の命令トレースデータを前記端子信号情報に同期させて記録する機能をさらに有することを特徴とする請求項 1 , 2 , 3 , 4 , 5 , 6 , 7 , または 8 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 1 0】 前記端子信号の再生速度を可変する機能をさらに有することを特徴とする請求項 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , または 9 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 1 1】 多入力 O R 素子に複数のロジック L S I の比較信号を接続して同時に複数のロジック L S I の解析を行う機能をさらに有することを特徴と

する請求項 1, 2, 3, 4, 5, 6, 7, 8, 9, または 10 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 1 2】 前記端子信号情報が、アナログ信号情報も含むことを特徴とする請求項 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, または 11 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 1 3】 解析対象ロジック L S I と正常なロジック L S I のアナログ／デジタル差分特性、またはデジタル／アナログ差分特性をあらかじめ取得し、アナログ変換特性を補正する機能をさらに有することを特徴とする請求項 1 2 に記載のロジック L S I の不良解析システム。

【請求項 1 4】 前記ロジック L S I は、オンチップデバッガを搭載していることを特徴とする請求項 1 2 または 1 3 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 1 5】 階層化したソフトウェアを有することを特徴とする請求項 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, または 14 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 1 6】 解析対象が、ロジック L S I を搭載したシステムであることを特徴とする請求項 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, または 15 のうちのいずれか 1 項に記載のロジック L S I の不良解析システム。

【請求項 1 7】 ソフトウェアを内蔵したロジック L S I の不良解析方法であって、

クロックに同期して前記ロジック L S I の端子信号情報を記録する手段と、

前記記録した端子信号情報をクロックに同期して再生する手段と、

前記再生した端子信号情報と、正常なロジック L S I の端子信号情報とを比較する手段と、

を含むことを特徴とするロジック L S I の不良解析方法。

【請求項 1 8】 ソフトウェアを内蔵したロジック L S I の不良解析方法であって、

クロックに同期して解析対象のロジック L S I の端子信号情報を記録する手段

と、

クロックに同期して正常なロジック L S I の端子信号情報を記録する手段と、  
前記記録した解析対象のロジック L S I の端子信号情報と、前記記録した正常なロジック L S I の端子信号情報とを比較する手段と、  
を含むことを特徴とするロジック L S I の不良解析方法。

【請求項 1 9】 ソフトウェアを内蔵したロジック L S I の不良解析方法であって、

クロックに同期して解析対象のロジック L S I の端子信号情報を記録する手段と、

前記記録した端子信号情報をクロックに同期して再生する手段と、  
前記再生した解析対象のロジック L S I の端子信号情報と、ロジック L S I のエミュレータの端子信号情報とを比較する手段と、  
を含むことを特徴とするロジック L S I の不良解析方法。

【請求項 2 0】 ソフトウェアを内蔵したロジック L S I の不良解析方法であって、

クロックに同期して解析対象のロジック L S I の端子信号情報を記録する手段と、

前記記録した解析対象のロジック L S I の端子信号情報と、ロジック L S I のエミュレータの端子信号情報とを比較する手段と、  
を含むことを特徴とするロジック L S I の不良解析方法。

【請求項 2 1】 前記端子信号情報は、特定期間におけるレジスタデータと RAM データの状態変化のトレースデータマップを含むことを特徴とする請求項 1 7, 1 8, 1 9, または 2 0 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【請求項 2 2】 解析対象のロジック L S I のトレースデータマップと、正常なロジック L S I のトレースデータマップとのトレース差分マップを生成する手段をさらに含むことを特徴とする請求項 1 7, 1 8, 1 9, 2 0, または 2 1 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【請求項 2 3】 複数の不具合を発生させて、前記トレース差分マップを複



数生成し、差分の平均とばらつきを統計処理して求める手段をさらに含むことを特徴とする請求項 1 7, 1 8, 1 9, 2 0, 2 1, または 2 2 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【請求項 2 4】 不具合発生前の R A M 及びレジスタの状態を保存し、不具合発生前から復活させる手段をさらに含むことを特徴とする請求項 1 7, 1 8, 1 9, 2 0, 2 1, 2 2, または 2 3 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【請求項 2 5】 C P U の命令トレースデータを前記端子信号情報に同期させて記録する手段をさらに含むことを特徴とする請求項 1 7, 1 8, 1 9, 2 0, 2 1, 2 2, 2 3, または 2 4 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【請求項 2 6】 前記端子信号の再生速度を可変する手段をさらに含むことを特徴とする請求項 1 7, 1 8, 1 9, 2 0, 2 1, 2 2, 2 3, 2 4, または 2 5 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【請求項 2 7】 多入力 O R 素子に複数のロジック L S I の比較信号を接続して同時に複数のロジック L S I の解析を行う手段をさらに含むことを特徴とする請求項 1 7, 1 8, 1 9, 2 0, 2 1, 2 2, 2 3, 2 4, 2 5, または 2 6 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【請求項 2 8】 前記端子信号情報が、アナログ信号情報も含むことを特徴とする請求項 1 7, 1 8, 1 9, 2 0, 2 1, 2 2, 2 3, 2 4, 2 5, 2 6, または 2 7 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【請求項 2 9】 解析対象ロジック L S I と正常なロジック L S I のアナログ／デジタル差分特性、またはデジタル／アナログ差分特性をあらかじめ取得し、アナログ変換特性を補正する手段をさらに含むことを特徴とする請求項 2 8 に記載のロジック L S I の不良解析方法。

【請求項 3 0】 前記ロジック L S I は、オンチップデバッガを搭載していることを特徴とする請求項 2 8 または 2 9 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【請求項 3 1】 解析対象が、ロジック L S I を搭載したシステムであるこ

とを特徴とする請求項 1 7, 1 8, 1 9, 2 0, 2 1, 2 2, 2 3, 2 4, 2 5, 2 6, 2 7, 2 8, 2 9, または 3 0 のうちのいずれか 1 項に記載のロジック L S I の不良解析方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

この発明は、ソフトウェアを組み込んだロジック L S I の不良解析システムに関する。

【0 0 0 2】

【従来の技術】

マイクロコンピュータや D S P (Digital Signal Processor) などの L S I (Large Scale Integration circuit) は、内部にソフトウェアを組み込んでいるロジック L S I である。以下、これを単に「マイコン」という。マイコンを使用したシステムでは、ウェハプロセス過程で発生する欠陥等により、出荷検査では検出できない不具合が発生することがある。以下、このような不具合があるマイコンを「不具合マイコン」といい、不具合のないマイコンを「正常マイコン」という。

【0 0 0 3】

図 5 3 に、不良解析の対象システム 1 に、不具合マイコン 5 が組み込まれている状態を模式的に示す。不具合マイコン 5 は、内部回路 4 を介して表示回路 2 と入出力回路 3 とに信号線で接続され、一方でインターフェイス回路 6 に信号線で接続される。図 5 3 中の矢印は、信号（情報）の出入りを示す（以下、本明細書、図面において同じ）。不具合マイコン 5 の不具合は、対象システム 1 に組み込まれてシステムが検査される時、若しくは対象システム 1 が使用される時に発見される。

【0 0 0 4】

従来、マイコンの不良解析は、マイコンを組み込んだシステムにて不具合を再現させ、このシステムを構成する回路信号の解析により原因を求めていた。例えば、マイコンの不具合が判明した場合に、その不具合マイコンの不良解析は、シ

ステムよりその不具合マイコンを分離し、L S I テスタにより詳細なテストプログラムを実行して行っていた。

【 0 0 0 5 】

上記従来技術に関連する技術文献としては次のものがある。

【 0 0 0 6 】

【特許文献 1】

特開平 6 - 9 5 9 1 3

【特許文献 2】

特開平 5 - 3 3 4 1 2 0

【特許文献 3】

特開 2 0 0 1 - 2 4 9 8 2 3

【 0 0 0 7 】

【発明が解決しようとする課題】

しかしながら不具合マイコンは、出荷検査項目で漏れた、特殊な条件が成立した場合でしか不具合が発生しないため、例え詳細なテストプログラムであっても、全ての不具合発生の可能性を検証することは、まず不可能であった。

【 0 0 0 8 】

例えば、マイコンをシステムに実装させた状態で人為的に不具合を発生させ、ロジックアナライザで信号解析する従来の不具合解析方法では、システムの起動から不具合を発生させるまでの条件が、マイコンのクロック分解能レベルで同一でない。このため、ソフトウェアの有する変数の状態が異なってしまい、不具合の発生確率、不具合の状況の再現性が悪かった。

【 0 0 0 9 】

また、マイコン単体をテストに装着し、テストプログラムを動作させる従来の不具合解析方法では、テストプログラムによりマイコンに与える信号中に不具合要因が含まれなかった場合に不具合は再現されなかった。またこのテストプログラムは、前述の人為的に不具合現象を発生させ、ロジックアナライザで信号解析する解析方法により不具合の信号を予想して作成される。このため、テストプログラムの作成には膨大な工数がかかり、しかも不具合にヒットする確率は低く、

効率が悪かった。

#### 【 0 0 1 0 】

以下に、マイコンの不具合発生原因と、不具合検知に至るまでの過程を、より詳細に説明する。

#### 【 0 0 1 1 】

図 5 4 は、正常マイコンの内部構成図である。図 5 4 に示すように、マイコン 7 は、ROM (Read-Only Memory) 8, CPU 1 0, RAM (Random Access Memory) 1 1, インターフェイスレジスタ 1 2, ペリフェラルレジスタ 1 3, I O (Input Output) ポート等の入出力インターフェイス 1 4, ペリフェラル 1 5, 入力端子 1 6, 出力端子 1 7 をもつ。そして、ROM 8 内にはアプリケーションソフトウェア 9 が存在する。図 5 4 中の矢印は、信号 (情報) の流れを示す。マイコン 7 では、アプリケーションソフトウェア 9 により RAM 1 1, レジスタ 1 2, 1 3 の書込みが行われ、ハードウェア (ROM 8, CPU 1 0, RAM 1 1, インターフェイス 1 4, ペリフェラル 1 5) が制御される。

#### 【 0 0 1 2 】

図 5 5 は、不具合マイコンの内部構成図である。図 5 5 を用いて、ハードウェアに何らかの不具合が発生した場合について説明する。図 5 5 中の☆印は、「不具合」, 「異常」, または「誤り」を示す。図 5 5 で、ペリフェラル 1 5 に不具合が発生していると仮定すると、不具合が発生する条件が成立することにより、ペリフェラル状態を示すペリフェラルレジスタ 1 3 の内容が「異常」となる。アプリケーションソフトウェア 9 がこのレジスタ 1 3 を読み、判断を誤る。そして、誤った情報が RAM 1 1 に書き込まれる。さらに、アプリケーションソフトウェア 9 はこの誤った情報が書き込まれた RAM 1 1 を読む。その値をポートより出力する処理に到達した時に、誤った値がインターフェイスレジスタ 1 2 に書き込まれ、インターフェイス 1 4 より誤った情報が出力される。不具合マイコン 5 の外部からは、この誤った情報が出力されて初めて不具合が検知できるようになる。

#### 【 0 0 1 3 】

以上説明したように、アプリケーションソフトウェアが介在するマイコンにお

いては、不具合が発生しても、発生から不具合が検知されるまでの間にタイムラグが生じる。このため不具合要因を解析するためには、検知された不具合の情報を元にアプリケーションソフトウェアの動作を予想し、RAM、レジスタ等の状態を予想した上で真の原因を推測する必要がある。

【 0 0 1 4 】

しかし、従来の解析方法では、以下のような問題点があった。

1. 不具合の再現性が悪い。
2. 真の不具合と、検出された不具合との関連性の解析が困難である。
3. 不具合の再発を防止するテスト方法の確立が困難である。
4. 解析システムの構築が困難である。

【 0 0 1 5 】

そこで本発明は、上記 1 ～ 4 の問題を解決し、不具合の再現性が良い、真の不具合と、検出された不具合との関連性の解析が容易である、不具合の再発を防止するテスト方法の確立が容易である、解析システムの構築が容易である、ロジック L S I の不良解析システム、及び不良解析方法を提供することを目的とする。

【 0 0 1 6 】

【課題を解決するための手段】

上記課題を解決するための本発明の不良解析システムは、ソフトウェアを内蔵したロジック L S I の不良解析システムであって、クロックに同期してロジック L S I の端子信号情報を記録する機能と、記録した端子信号情報をクロックに同期して再生する機能と、再生した端子信号情報を正常なロジック L S I の端子信号情報とを比較する機能とを有することを特徴とする。このような構成により、不具合の再現性が良くなる。

【 0 0 1 7 】

また、上記課題を解決するための本発明の他の不良解析システムは、クロックに同期して解析対象のロジック L S I の端子信号情報を記録する機能と、クロックに同期して正常なロジック L S I の端子信号情報を記録する機能と、記録した解析対象のロジック L S I の端子信号情報と記録した正常なロジック L S I の端子信号情報とを比較する機能とを有することを特徴とする。

【 0 0 1 8 】

また、上記課題を解決するための本発明の他の不良解析システムは、クロックに同期して解析対象のロジック L S I の端子信号情報を記録する機能と、記録した端子信号情報をクロックに同期して再生する機能と、再生した解析対象のロジック L S I の端子信号情報とロジック L S I のエミュレータの端子信号情報とを比較する機能とを有することを特徴とする。

【 0 0 1 9 】

また、上記課題を解決するための本発明の他の不良解析システムは、クロックに同期して解析対象のロジック L S I の端子信号情報を記録する機能と、記録した解析対象のロジック L S I の端子信号情報とロジック L S I のエミュレータの端子信号情報とを比較する機能とを有することを特徴とする。

【 0 0 2 0 】

また、上記課題を解決するための本発明の他の不良解析システムは、上記いずれかの構成で、端子信号情報は、特定期間におけるレジスタデータと R A M データの状態変化のトレースデータマップを含むことを特徴とする。

【 0 0 2 1 】

また、上記課題を解決するための本発明の他の不良解析システムは、上記いずれかの構成で、解析対象のロジック L S I のトレースデータマップと正常なロジック L S I のトレースデータマップとのトレース差分マップを生成する機能をさらに有することを特徴とする。

【 0 0 2 2 】

また、上記課題を解決するための本発明の他の不良解析システムは、上記いずれかの構成で、複数の不具合を発生させて、トレース差分マップを複数生成し、差分の平均とばらつきを統計処理して求める機能をさらに有することを特徴とする。

【 0 0 2 3 】

また、上記課題を解決するための本発明の他の不良解析システムは、上記いずれかの構成で、不具合発生前の R A M 及びレジスタの状態を保存し、不具合発生前から復活させる機能をさらに有することを特徴とする。

【 0 0 2 4 】

また、上記課題を解決するための本発明の他の不良解析システムは、上記いずれかの構成で、CPUの命令トレースデータを端子信号情報に同期させて記録する機能をさらに有することを特徴とする。

【 0 0 2 5 】

また、上記課題を解決するための本発明の他の不良解析システムは、上記いずれかの構成で、端子信号の再生速度を可変する機能をさらに有することを特徴とする。

【 0 0 2 6 】

また、上記課題を解決するための本発明の他の不良解析システムは、上記いずれかの構成で、多入力OR素子に複数のロジックLSIの比較信号を接続して同時に複数のロジックLSIの解析を行う機能をさらに有することを特徴とする。

【 0 0 2 7 】

また、上記課題を解決するための本発明の他の不良解析システムは、上記いずれかの構成で、端子信号情報が、アナログ信号情報も含むことを特徴とする。またさらに、解析対象ロジックLSIと正常なロジックLSIのアナログ／デジタル差分特性、またはデジタル／アナログ差分特性をあらかじめ取得し、アナログ変換特性を補正する機能をさらに有していてもよい。

【 0 0 2 8 】

また、ロジックLSIは、オンチップデバッガを搭載していてもよい。

【 0 0 2 9 】

なお、解析対象はロジックLSIに限定されず、ロジックLSIを搭載したシステムであってもよい。

【 0 0 3 0 】

また、上記課題を解決するための本発明のソフトウェアを内蔵したロジックLSIの不良解析方法は、クロックに同期してロジックLSIの端子信号情報を記録する手段と、記録した端子信号情報をクロックに同期して再生する手段と、再生した端子信号情報を、正常なロジックLSIの端子信号情報と比較する手段とを含むことを特徴とする。

【 0 0 3 1 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、クロックに同期して解析対象のロジック L S I の端子信号情報を記録する手段と、クロックに同期して正常なロジック L S I の端子信号情報を記録する手段と、記録した解析対象のロジック L S I の端子信号情報と記録した正常なロジック L S I の端子信号情報とを比較する手段とを含むことを特徴とする。

【 0 0 3 2 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、クロックに同期して解析対象のロジック L S I の端子信号情報を記録する手段と、記録した端子信号情報をクロックに同期して再生する手段と、再生した解析対象のロジック L S I の端子信号情報とロジック L S I のエミュレータの端子信号情報とを比較する手段とを含むことを特徴とする。

【 0 0 3 3 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、クロックに同期して解析対象のロジック L S I の端子信号情報を記録する手段と、記録した解析対象のロジック L S I の端子信号情報とロジック L S I のエミュレータの端子信号情報とを比較する手段とを含むことを特徴とする。

【 0 0 3 4 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、上記いずれかの構成において、端子信号情報は、特定期間におけるレジスタデータと R A M データの状態変化のトレースデータマップを含むことを特徴とする。

【 0 0 3 5 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、上記いずれかの構成において、解析対象のロジック L S I のトレースデータマップと、正常なロジック L S I のトレースデータマップとのトレース差分マップを生成する手段をさらに含むことを特徴とする。

【 0 0 3 6 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法



は、上記いずれかの構成において、複数の不具合を発生させて、トレース差分マップを複数生成し、差分の平均とばらつきを統計処理して求める手段をさらに含むことを特徴とする。

【 0 0 3 7 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、上記いずれかの構成において、不具合発生前の R A M 及びレジスタの状態を保存し、不具合発生前から復活させる手段をさらに含むことを特徴とする。

【 0 0 3 8 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、上記いずれかの構成において、 C P U の命令トレースデータを端子信号情報に同期させて記録する手段をさらに含むことを特徴とする。

【 0 0 3 9 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、上記いずれかの構成において、端子信号の再生速度を可変する手段をさらに含むことを特徴とする。

【 0 0 4 0 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、上記いずれかの構成において、多入力 O R 素子に複数のロジック L S I の比較信号を接続して同時に複数のロジック L S I の解析を行う手段をさらに含むことを特徴とする。

【 0 0 4 1 】

また、上記課題を解決するための本発明の他のロジック L S I の不良解析方法は、上記いずれかの構成において、端子信号情報が、アナログ信号情報も含むことを特徴とする。またさらに、解析対象ロジック L S I と正常なロジック L S I のアナログ／デジタル差分特性、またはデジタル／アナログ差分特性をあらかじめ取得し、アナログ変換特性を補正する手段をさらに含んでも良い。

【 0 0 4 2 】

また、上記いずれかの不良解析方法において、ロジック L S I は、オンチップデバッガを搭載していてもよい。

【0043】

また、本発明の不良解析方法において、解析対象はロジック L S I に限定されず、ロジック L S I を搭載したシステムであってもよい。

【0044】

【実施例】

以下に、本発明のいくつかの実施例を、図面を用いて説明する。なお、本明細書及び図面において、実質的に同一の機能構成を有する構成要素については、同一の符号を付することにより重複説明を省略する。

【0045】

(実施例 1)

実施例 1 は、1. 不具合の再現性が悪い、という問題を解決している。この問題を解決し、不具合を確実に再現するために、実施例 1 の不良解析システムではマイコンの端子状態をマイコンのクロックに同期して記録し、端子トレースデータとして保存する。そして、この端子トレースデータをクロックに同期してマイコンに注入し、記録した端子トレースデータの再生を行う。

【0046】

図 1 に、実施例 1 による不良解析システムの構成を示す。このシステムは、制御を行うワークステーション 26、及び解析ボード 24 を含んで構成される。対象システム 21 に装着されたままの不具合マイコン 22 は、全ての端子、若しくは VDD、GND の除く残りの端子が、検査プローブ 23 を介して解析ボード 24 に接続される。電源の VDD 端子や接地の GND 端子は、信号端子に比べて不具合が現れる確率が低いので、適用外としてもかまわない。解析ボード 24 とワークステーション 26 とは、接続ハーネス 25 内の各種信号線で接続される。

【0047】

図 2 は、実施例 1 のワークステーション 26 の構成図である。ワークステーション 26 は、CPU 27、ストレージ 28、クロック制御ボード 32、端子デジタル I/O ボード 33、制御デジタル I/O ボード 34 を含んで構成される。CPU 27 は、ワークステーション 26 の制御の中枢部であり、ストレージ 28 は、データを保存するための大容量データストレージシステムである。クロック制御ボ

ード 3 2 は、ソフトウェアの制御により任意のクロック周波数の発振を行い、外部に送出する機能と、外部から注入されたクロックに対して搭載された他のボードの同期を行う機能を有する。端子デジタル I O ボード 3 3 は、マイコン 2 2 の端子以上のチャンネル数をもった入力／出力の切替可能な双方向デジタル I O ボードである。制御デジタル I O ボード 3 4 は、解析ボード 2 4 上のマイコンインターフェイスモジュール 3 5 を制御する機能を有する。クロック入力線は、クロック信号の入力線であり、端子信号線は、マイコン 2 2 の端子と端子デジタル I O ボード 3 3 とを接続する信号線であり、解析ボード制御線は、マイコン 2 2 のインターフェイスモジュール 3 5 と制御デジタル I O ボード 3 4 とを接続する信号線である。接続ハーネス 2 5 は、上記クロック入力線、端子信号線、及び解析ボード制御線を束ね、解析ボード 2 4 とワークステーション 2 6 とを接続する。

## 【 0 0 4 8 】

図 3 は、実施例 1 の解析ボード 2 4 の構成図である。図 3 に示すように、解析ボード 2 4 は、解析ボード 2 4 上にマイコン 2 2 を装着するためのマイコンソケット 3 6 と、対象システム 2 1 に搭載されている不具合マイコン 2 2 の端子と解析ボード 2 4 上に搭載されるマイコンソケット 3 6 の端子とを接続する検査プローブ 2 3 と、マイコンインターフェイスモジュール 3 5 とを含んでいる。マイコンインターフェイスモジュール 3 5 は、マイコンソケット 3 6 の端子信号と、ワークステーション 2 6 から接続ハーネス 2 5 を介して送出される信号との制御、調停を行うように構成される。

## 【 0 0 4 9 】

図 4 は、実施例 1 のマイコンインターフェイスモジュール 3 5 の内部構成図である。図 4 に示すように、マイコンインターフェイスモジュール 3 5 の内部は、大きく次のブロックに分けられる。制御ブロック 2 0 は、ワークステーション 2 6 より制御デジタル I O ボード 3 4 の制御信号を受信し、信号を解析する。クロックブロック 3 7 は、マイコンソケットのクロック端子に対するクロック信号の入出力を制御し、同期信号を生成する。端子制御ブロック 3 8 (複数) は、マイコン端子毎に存在し、マイコン端子毎の制御を行う。端子信号差異検出ブロック 3 9 は、端子とデータとの差異を検出する。ワークステーション 2 6 側の信号を

制御信号デコーダ 4 1 が受け、解釈し、必要に応じて端子切断制御部 4 0 を制御し、端子制御ブロック 3 8 の各々に対し、マイコン 2 2 側の信号の接続／切断を切り替える信号を送り出す。クロック方向制御部 4 2 は、制御信号デコーダ 4 1 からの制御信号を受け取り、クロックブロック 3 7 に対するクロック信号の方向制御信号を送り出す。

## 【 0 0 5 0 】

クロック端子からマイコンクロック信号線を介して入力したクロック信号は、クロック入力バッファ 4 3 によりバッファリングされ、クロック入力信号としてワークステーション 2 6 側のクロック入力ポートに、クロック入力線を介して送出される。また、ワークステーション 2 6 側のクロック出力ポートより出力されたクロック出力信号はクロック出力バッファ 4 4 で受信されて、マイコンソケット 3 6 のクロック端子に出力される。

## 【 0 0 5 1 】

クロック入力バッファ 4 3 とクロック出力バッファ 4 4 は、クロック信号の方向制御信号により排他的に動作する。本実施例では、クロック信号の方向制御信号が” L ” のとき信号方向が「マイコン」→「ワークステーション」で、” H ” のとき信号方向が「ワークステーション」→「マイコン」となる。

## 【 0 0 5 2 】

マイコン端子信号はバッファリングされ、ワークステーション 2 6 に向けて送出されると同時に、端子制御ブロックの同期信号として各端子制御ブロック 3 8 に送出される。

## 【 0 0 5 3 】

マイコン端子毎に存在する端子制御ブロック 3 8 は、マイコンソケット 3 6 の端子より得られるマイコン端子信号と、ワークステーション 2 6 より送出される端子信号の接続／切断の制御を行う。信号の接続／切断は、アナログスイッチで行う。端子制御ブロック 3 8 は、マイコン端子信号とワークステーション 2 6 より送出される端子信号の比較を、クロックブロック 3 7 からの同期信号に同期して行い、信号間に差異が発生した場合は差異信号を” H ” として出力する。この差異の検出は、排他論理和 X O R 回路と D - ラッチとの組み合わせで行う。

【 0 0 5 4 】

アナログスイッチが導通している状態では差異が発生しないため、常に差異信号は” L ” が出力される。端子信号差異検出ブロック 3 9 は、全端子制御ブロック 3 8 より出力された差異信号の論理和（ O R ）をマルチゲート O R で取り、差異信号としてワークステーション 2 6 へ出力する。

【 0 0 5 5 】

ワークステーション 2 6 と解析ボード 2 4 とは、接続ハーネス 2 5 で束ねられて、以下の向きと本数の信号線が接続される。端子信号線は、解析ボード 2 4 とワークステーション 2 6 と双方向で、有効端子数分である。クロック入力線は、ワークステーション 2 6 から解析ボード 2 4 へ向かい、 1 本である。クロック出力線は、解析ボード 2 4 からワークステーション 2 6 へ向かい、 1 本、差異信号線も、解析ボード 2 4 からワークステーション 2 6 へ向かい、 1 本である。プローブ制御線は、双方向で必要数分である。

【 0 0 5 6 】

図 2，図 3 を用いて、記録時の信号の流れを説明すると、不具合マイコン 2 2 のクロック端子→検査プローブ 2 3 →解析ボード 2 4 →クロック入力線→クロック制御ボード 3 2 と、不具合マイコン 2 2 の端子→検査プローブ 2 3 →解析ボード 2 4 →クロック入力線→端子デジタル I O ボード 3 3 となる。

【 0 0 5 7 】

記録時にはワークステーション 2 6 のストレージ 2 8 に、図 5 に示すような、クロック数 0 から終了点までの、端子毎の” H ”，” L ” 状態（または、 1， 0 状態）を示す、端子トレースデータ 3 1 が記録される。

【 0 0 5 8 】

次に、再生時について説明する。図 6 は、実施例 1 における、再生時のワークステーション 2 6 の構成図である。記録時とは次の点で異なる。再生時はクロック入力線が無効、クロック出力線が有効となり、ワークステーション 2 6 よりクロック信号が送出される。端子信号線は、信号の方向は、ワークステーション 2 6 から解析ボード 2 4 へ方向に反転する。差異信号線は、記録時は無効であったのが有効となり、差異信号は、制御デジタル I O ボード 3 4 の入力ポートに入

力される。

【0059】

図7は、実施例1における、再生時の解析ボード24の構成図である。図7に示すように、再生時は検査プローブ23をはずし、解析ボード24上のマイコンソケット36に、正常マイコン45を装着する。記録時とは次の点で異なる。前述のように、再生時はクロック入力線が無効、クロック出力線が有効となり、ワークステーション26よりクロック信号が流入する。端子信号線は、信号の方向が、ワークステーション26から解析ボード24へ方向に反転する。差異信号線は、記録時は無効であったが有効となり、解析ボード24より送出される。

【0060】

図8は、実施例1における、再生時のマイコンインターフェイスモジュール35の構成図である。図8に示すように、解析ボード24の制御信号が変更されるため、記録時に比べて内部状態が変化する。記録時と次の点が異なる。クロック方向制御信号は”H”となり、ワークステーション26で生成され、クロック出力線を介して流入したクロック信号を、正常マイコン45に供給する。正常マイコン45の出力端子に関しては、切断制御信号は”L”となり、端子制御ブロック38において、アナログスイッチが切断される。正常マイコン45側の端子信号と、ワークステーション26側の端子信号が異なった場合、ラッチシンクロ信号に同期して、差異信号が”H”に変化する。そして、端子信号差異検出ブロック39により、差異信号線が”H”に変化する。

【0061】

再生時にはワークステーション26のストレージ28に、図9に示すような、端子毎の、クロック数0から終了点までの、正常マイコン出力と不具合マイコン出力の端子トレースデータ31が記録される。

【0062】

次に、実施例1の動作を説明する。本実施例のシステムは、不具合発生状態の記録と、不具合発生状態の再現の、2段階の動作を行う。

【0063】

記録時には、ワークステーション26において記録ソフトウェア29がCPU

27にロードされ、実行される。記録ソフトウェア29は以下の操作を行い、記録スタートを待つ。まず、マイコンインターフェイスモジュール35の設定をセットアップするように制御デジタルIOボード34を動作させ、解析ボード24の制御線を介してマイコンインターフェイスモジュール35に設定信号を送出する。次に、端子デジタルIOボード33の全ポートを入力に設定する。次に、クロック制御ボード32のクロック入力を有効にし、クロック入力線より流入したクロック信号に対し、端子デジタルIOボード33を同期動作させるように設定する。次に、クロック信号が入力に同期して、端子デジタルIOボード33の入力状態を、ストレージ上の端子トレースデータとして順次保存するよう準備する。

#### 【0064】

解析ボード24は、解析ボード制御信号に従い、制御ブロック20が次のセットアップを行う。まず、クロック信号の方向制御信号を”L”に設定し、クロックブロック37が、不具合マイコン22からワークステーション26に向かう方向にクロック信号を流すように設定する。次に、全ての端子制御ブロック38のアナログスイッチを接続し、不具合マイコン22の端子と端子デジタルIOボード33のポートとを接続する。次に、不具合の発生している対象システム21を起動、操作し、不具合を発生させる。

#### 【0065】

操作者が不具合の発生を確認した後、ワークステーション26を操作し、記録を完了させる。操作者は、異常表示、動作異常、検査装置のアラーム等により不具合を判断する。この時点でデータストレージシステム28は、不具合発生マイコン22の起動から停止までの全端子信号が端子トレースデータ31として記録されている(図5)。

#### 【0066】

上記の記録ソフトウェア29の動作を、図10に示す。S11で端子デジタルIOボード33、制御デジタルIOボード34を初期化し(S11)、S12で解析ボード24のマイコンインターフェイスモジュール35のクロック方向、アナログスイッチ設定を初期化し(S12)、S13で読み出しカウンタを初期化

する（S 1 3）。そして、S 1 4 でクロック入力が発生するまで待つ（S 1 4）。次に、S 1 5 でクロックに同期して端子状態を端子デジタル I O ボード 3 3 でラッチする（S 1 5）。次に、S 1 6 で、端子デジタル I O ボード 3 3 でラッチした端子トレースデータをストレージ 2 8 に保存する（S 1 6）。S 1 7 で、操作者が不具合を認識したら停止し、認識しなければ、S 1 8 で読み出しカウンタをインクリメントし、S 1 5 へ戻り（S 1 8）、不具合を認識するまで記録を継続する。

## 【 0 0 6 7 】

再生時には、解析ボード 2 4 から検査プローブ 2 3 をはずし、正常マイコン 4 5 をマイコンソケット 3 6 に装着する。次に、ワークステーション 2 6 において再生ソフトウェア 3 0 がロードされ、実行される。再生ソフトウェア 3 0 は、以下の操作を行い、不具合の検出を待つ。まず、端子デジタル I O ボード 3 3 の全ポートを、出力に設定する。次に、マイコンインターフェイスモジュール 3 5 の設定をセットアップするように制御デジタル I O ボード 3 4 を動作させ、解析ボード制御線を介してマイコンインターフェイスモジュール 3 5 に設定信号を送出する。

## 【 0 0 6 8 】

次に、ストレージ 2 8 上の端子トレースデータ 3 1 を、クロック信号に同期して、端子デジタル I O ボード 3 3 から順次送出するようにセットアップする。次に、クロック制御ボード 3 2 よりクロック出力を開始し、再生をスタートさせる。再生中は、マイコンインターフェイスモジュール 3 5 において、マイコン出力端子の信号に関して実際の信号レベルと端子トレースデータ 3 1 に対応した端子信号の電圧の比較を随時行う。そして、信号に差異を生じた場合は、差異信号線が” L ” から” H ” へと変化し、これが制御デジタル I O ボード 3 4 で検知され、クロック送出を停止し、再生動作を停止する。

## 【 0 0 6 9 】

上記の再生ソフトウェア 3 0 の動作を、図 1 1 に示す。S 2 1 で端子デジタル I O ボード 3 3、制御デジタル I O ボード 3 4 を初期化し（S 2 1）、S 2 2 で解析ボード 2 4 のマイコンインターフェイスモジュール 3 5 のクロック方向、ア



ナログスイッチ設定を初期化し（S 2 2），S 2 3で読み出しカウンタを初期化する（S 2 3）。そして，S 2 4でクロック発振を開始する（S 2 4）。次に，S 2 5でストレージ28より端子トレースデータ31を端子デジタルI Oボード33にロードする（S 2 5）。次に，S 2 6で，クロック信号に同期して，端子デジタルI Oボード33より端子信号を送出する（S 2 6）。S 2 7で，差異信号があれば発振を停止し（S 2 8），なければ，S 2 8で読み出しカウンタをインクリメントして（S 2 9），S 2 5へ戻り，差異信号があるまで再生処理を継続する。

## 【 0 0 7 0 】

以上示した構成であるので，実施例1によれば，不具合の再現性が良くなる。

## 【 0 0 7 1 】

## （実施例2）

実施例2は，1．不具合の再現性が悪い，という問題を解決している。実施例2では，実施例1のように解析ボード24上にマイコンインターフェイスモジュール35を必要としない，簡素化した解析ボード24による不具合マイコン解析システムを提供する。

## 【 0 0 7 2 】

図12は，実施例2における，記録時のワークステーション26の構成図である。実施例1からの変更点は，制御デジタルI Oボード34，解析ボード制御線がないことである。これらはマイコンインターフェイスモジュール35を使用しないため，必要としない。

## 【 0 0 7 3 】

図13は，実施例2における記録時の解析ボード24の構成図である。実施例1からの変更点は，マイコンインターフェイスモジュールを使用しないことである。このため，各端子信号線は，直接マイコンソケット36に接続される。

## 【 0 0 7 4 】

図14は，実施例2における再生時のワークステーション26の構成図である。実施例1からの変更点は，制御デジタルI Oボード34，解析ボード制御線がないことである。これらは，マイコンインターフェイスモジュール35を使用し

ないため、必要としない。端子トレースデータ 3 1 は、再生時にマイコンの出力端子の状態を記録したデータである。

## 【 0 0 7 5 】

図 1 5 は、実施例 2 における再生時の解析ボード 2 4 の構成図である。実施例 1 からの変更点は、マイコンインターフェイスモジュール 3 5 を使用しないことである。このため各端子信号線は、直接マイコンソケット 3 6 に接続される。また、端子信号線の方法は、次のようになる。入力端子は、ワークステーション 2 6 から正常マイコン 4 5 へ、出力端子は、正常マイコン 4 5 からワークステーション 2 6 へと向かう。

## 【 0 0 7 6 】

実施例 2 のシステムの動作は、不具合データの記録時は実施例 1 とほぼ同様である。しかし、マイコンインターフェイスモジュール 3 5 が存在しないため、一部が異なる。まず、マイコンインターフェイスモジュール無し対応の記録ソフトウェア 2 9' をロードする。制御デジタル I O ボード 3 4 が存在しないため、解析ボード 2 4 の初期設定が不要である。次に、実施例 1 と同様に記録を行い、端子トレースデータ 3 1 を得る。

## 【 0 0 7 7 】

再生は、まず、マイコンインターフェイスモジュール無し対応の再生ソフトウェア 3 0' をロードする。マイコン端子が「入力」の場合、実施例 1 と同様に端子トレースデータ 3 1 を出力する。マイコン端子が「出力」の場合、端子デジタル I O ボード 3 3 の方向設定を、正常マイコン 4 5 からワークステーション 2 6 へとし、端子デジタル I O ボード 3 3 への入力信号を、記録時と同じように端子トレースデータ 3 1' に保存する。再生動作は、実施例 1 のように終点検出を行わず、再生中の端子トレースデータ 3 1 が終了した時点で終了する。

## 【 0 0 7 8 】

差異点の検出に関して、図 1 6 の端子トレースデータ比較図を用いて説明する。記録、再生において、それぞれ端子トレースデータ（記録時） 3 1、及び端子トレースデータ（再生時） 3 1' の、2 つのデータが得られる。この 2 データをワークステーション 2 6 において比較し、端子トレースデータ差分として出力す

る。端子トレースデータ差分において、不具合マイコン 2 2 と正常マイコン 4 5 の差が明確に判別可能となり、実施例 1 と同様の不具合発生タイミング結果を得ることができる。

【 0 0 7 9 】

実施例 2 では、実施例 1 と同等の効果が、マイコンインターフェイスモジュール 3 5 および制御デジタル I O ボード 3 4 の無い簡素化したシステムで得ることができる。従って、システムが簡素化された分、コストダウンが可能になる。

【 0 0 8 0 】

(実施例 3)

実施例 3 は、1. 不具合の再現性が悪い、という問題と、2. 真の不具合と検出された不具合との関連性の解析が困難である、という問題を解決する。

【 0 0 8 1 】

実施例 3 の不具合データ記録時の構成は、実施例 1 と同様である。図 1 7 は、実施例 3 における、不具合データ再生時のシステム構成図である。図 1 7 に示すように、実施例 1 の正常マイコン 4 5 (図 7) を、インサーキットエミュレータ 4 6 に交換したものである。実施例 3 において、不具合データ記録時、再生時の動作は、実施例 1 と同様である。

【 0 0 8 2 】

実施例 3 では、エミュレータ 4 6 を使用することにより、不具合発生時点の正常マイコン 4 5 の内部状態を解析することが可能になる。また、この解析結果から原因を追及するために、不具合発生時点から遡った時点での内部状態を、次の手順で解析することができる。

【 0 0 8 3 】

まず、起動から不具合発生時点より前の任意の時点まで、クロック制御ボード 3 2 よりクロックを供給する。その後クロックを停止し、インサーキットエミュレータ 4 6 を停止し、実行モードよりモニタモードに変えて、マイコンの内部解析を行う。このようにすれば、不具合発生までの端子トレースデータ 3 1 が保存されているため、任意の時点まで動作させて解析することができる。

【 0 0 8 4 】

## (実施例 4)

実施例 4 は、 1. 不具合の再現性が悪い、という問題と、 2. 真の不具合と検出された不具合との関連性の解析が困難である、という問題を解決する。

## 【 0 0 8 5 】

実施例 3 では、インサーキットエミュレータ 4 6 を接続することで、正常マイコン 4 5 の内部情報が取得可能であるが、解析を行う上で不具合マイコン 2 2 の内部情報が取得可能であれば、より深い解析が可能になり、解析速度も向上する。実施例 4 では、デバッグモニタを搭載した不具合マイコン 2 2 において、内部情報を取得する。このために、実施例 4 では、評価対象のマイコンとして、デバッグモニタソフトウェアをあらかじめ搭載したマイコンを使用する。

## 【 0 0 8 6 】

図 1 8 は、実施例 4 における、マイコンの内部構成図である。図 1 8 に示すように、マイコン 5 0 の内部の ROM 5 1 上に、デバッグモニタソフトウェア 5 3 が搭載されているが、記録／再生時には動作しない。実施例 4 では、デバッグモニタ 5 3 はアプリケーションソフトウェア 5 2 で使用しないシリアルインターフェイス入力端子 6 4 に信号が入力された時点で起動する。

## 【 0 0 8 7 】

次に、実施例 4 における、解析時のマイコンの内部構成を図 1 9 に示す。アプリケーションソフトウェア 5 2 の動作中に、シリアル入力端子 6 4 に対して入力が発生した場合、動作がアプリケーションソフトウェア 5 2 からデバッグモニタ 5 3 に遷移し、デバッグモニタ 5 3 においてシリアルインターフェイス 6 0 を介してデバッグコマンドを受け付け、コマンドで指定された RAM 5 5、内部レジスタ情報をシリアルインターフェイス 6 0 のシリアル出力端子 6 5 よりレスポンスとして読み出すことが可能になる。

## 【 0 0 8 8 】

デバッグモニタ機能を動作させた時点で、アプリケーションソフトウェア 5.2 の動作は中断するが、マイコン 5 0 の外側の回路動作は中断しないため、デバッグモニタ 5 3 起動後は、マイコン 5 0 と外部回路との同期が取れず、起動後には正常に実行されない。

## 【 0 0 8 9 】

実施例 4 における不具合データ記録時、再生時の構成は、実施例 1 と同様である。

## 【 0 0 9 0 】

図 2 0 は、実施例 4 における不具合データ解析時の解析ボードの構成図である。実施例 1 において再生時は正常マイコン 4 5 を搭載したが、実施例 4 では不具合マイコン 2 2 の解析を目的とするため、対象システムより不具合の発生したマイコンを抜き取り、解析ボード 2 4 に装着する。実施例 4 では実施例 1 に対して、デバッグモニタ 5 3 を動作させるためのデバッグモニタ入力線、及びデバッグモニタ出力線が解析ボード 2 4 に追加されている。

## 【 0 0 9 1 】

次に、解析時のワークステーション 6 6 の構成に関して、図 2 1 で説明する。実施例 1 から変更のある構成要素を次に列挙する。解析ソフトウェア 6 9 は、マイコン 5 0 より、レジスタ状態、RAM 状態を取得するソフトウェアである。シリアル I O ボード 7 6 は、デバッグモニタ 5 3 を動作させるため、シリアル信号を送受信する。レジスタ状態データ 7 1 は、取得したレジスタのデータを保存するストレージ 6 8 上の領域である。RAM 状態データ 7 2 は、取得した RAM のデータを保存するストレージ 6 8 上の領域である。なお、簡便のために、このマイコンにおいてはメモリマップド I O とし、レジスタは RAM アドレス 0 ~ 0 F F H に配置されるものとする。なお、本明細書、図面において数字の後ろに付した「H」は、その数字が 1 6 進数であることを表す。

## 【 0 0 9 2 】

実施例 4 においては、記録時、再生時の動作は、実施例 1 と同様である。解析時には、解析ソフトウェア 6 9 を起動する。解析ソフトウェア 6 9 のフローチャートを図 2 2 に示す。

## 【 0 0 9 3 】

解析ソフトウェアがスタートすると、まず最初に、S 3 1 で RAM 5 5 の読み出しタイミング、読み出しアドレスを設定する (S 3 1) 。

## 【 0 0 9 4 】

次に、S 3 2 ~ S 3 7, S 4 1 で、設定されたタイミングまで再生動作させる再生処理を行う。再生処理は、再生と同じ動作をするが、設定した読み出しタイミングまで再生すると処理を抜ける。次に、S 3 8 で、デバッグモニタ 5 3 への R A M 読み出しコマンド（シリアル信号）をシリアル I O ボードにより送出し、不具合マイコンのシリアル端子に信号を入力し、デバッグモニタを起動させ、マイコンの処理をデバッグモニタに切り替える（S 3 8）。デバッグモニタが起動すると、コマンドを受け付け、指定された R A M データをレスポンスとしてシリアルインターフェイス出力から送出する。S 3 9 で読み出したデータをストレージ 6 8 に保存し（S 3 9）、S 4 0 で発振停止する（S 4 0）。

## 【 0 0 9 5 】

デバッグモニタ 5 3 の動作において、読み出しコマンド送出により、R A M 読み出しまでのタイムラグが生じるが、実施例 2 で示したインサーキットエミュレータ 4 6 により解析を行うことにより、デバッグモニタ動作時の内部動作が判別するため、コマンド送出後、R A M 読み出しまでの正確なタイミングをあらかじめ把握することができる。

## 【 0 0 9 6 】

以上示したように、実施例 4 では、不具合発生マイコンの R A M、レジスタの動作を把握するので、不具合解析が容易になる。

## 【 0 0 9 7 】

## （実施例 5）

実施例 5 は、1. 不具合の再現性が悪い、という問題と、2. 真の不具合と検出された不具合との関連性の解析が困難である、という問題を解決する。実施例 4 の場合でも、デバッグモニタ動作の 1 コマンド送出からレスポンス受信までに要するタイムラグが生じる。この間にペリフェラルは動作を継続することとなり、ペリフェラル関連レジスタが変化してしまう可能性がある。R A M 読み出しのアドレスを順次インクリメントし、R A M を順に読んだとしても、読み出し動作中に R A M、レジスタ内容が変化する可能性が高い。従って、デバッグモニタを起動した時点の R A M データと乖離している可能性があり、信頼性が低い。実施例 5 ではこの欠点を補い、解析に必要な R A M データを時間軸に沿って収集する

## 【 0 0 9 8 】

実施例 5 は基本的に実施例 4 と同じ方法で R A M データを取得するが、取得するデータをタイムラグの影響が無視可能な数に制限する。再生は何回でも再現可能であるため、R A M を取得する領域を順次スキャンし、必要な領域の R A M を複数回繰り返して取得する。特定のタイミングの R A M の取得が終了したら、次に取得するタイミングを変えて、時間軸方向に順次スキャンする。以上により、実施例 5 では、指定の R A M 領域、及び指定の時間範囲におけるレジスタ、R A M の変化データが得られる。

## 【 0 0 9 9 】

実施例 5 では、不具合データ記録時の構成は実施例 1 と同様である。また、不具合データ記録時の動作も実施例 1 と同様である。図 2 3 に、実施例 5 における、不具合データ解析時の構成図を示す。

## 【 0 1 0 0 】

ワークステーション 7 7 には、自動トレースソフトウェア 8 0 が動作する。ストレージ 7 9 には、記録時に生成した端子トレースデータ 8 1、レジストトレースデータ 8 2、R A M トレースデータ 8 3 が保存されている。

## 【 0 1 0 1 】

実施例 5 における、不具合データ解析時の動作を、自動トレースソフトウェアのフローチャート図 2 4 に示す。プログラムがスタートすると、S 4 1 で、読み込みタイミング範囲の設定、読み込みアドレスの範囲の設定を行い（S 4 1）、S 4 2 で読み出しアドレスを初期化し（S 4 2）、S 4 3 ～ S 4 8、S 5 4 で、実施例 4 と同様に、レジスタ、R A M の状態取得を行う。

## 【 0 1 0 2 】

この状態での端子トレースデータ 8 1、レジスタトレースデータ 8 2、R A M トレースデータ 8 3 の相関を、図 2 5 に示す。最初の 1 回は、「取得済み」で示される領域のレジストデータが取得され、レジストトレースデータに保存される。アドレス範囲を読み出し完了していないため、次の読み出しアドレス（「取得中」で示される領域）に設定され、再生処理→デバッグモニタによるレジスタ読

み出しを順次繰り返す。

【 0 1 0 3 】

指定アドレスを全て読み終えた場合、次の読み出しタイミングを初期値 S より + 1 インクリメントして設定し、同様にレジスタ、RAM のトレースデータの収集を継続する。

【 0 1 0 4 】

全ての読み出しを完了すると、図 2 6 に示す端子トレースデータ 8 1、レジスタトレースデータ 8 2、RAM トレースデータ 8 3 を得ることができる。

【 0 1 0 5 】

以上示したように、実施例 5 では、端子トレースデータ 8 1、レジスタトレースデータ 8 2、RAM トレースデータ 8 3 を解析することにより、時系列に沿って詳細に不具合解析作業が可能になる。

【 0 1 0 6 】

(実施例 6)

実施例 6 は、1. 不具合の再現性が悪い、という問題と、2. 真の不具合と検出された不具合との関連性の解析が困難である、という問題を解決する。実施例 5 によって、端子、レジスタ、RAM の各トレースデータの収集が可能となったが、不具合箇所を解析する必要がある。そこで実施例 6 では、トレース上での不具合箇所の明確化を目的とし、実施例 5 で取得したトレースデータに加え、正常マイコンに対しても同様のトレースを取得し、トレースデータ間の差分を得ることにより差異の発生する明確な箇所を判別する。

【 0 1 0 7 】

実施例 6 を、図 2 7 のトレース差分マップによって説明する。図 2 7 は、不具合マイコントレースマップと、正常マイコントレースマップと、両者の差分であるトレース差分マップにより構成される。

【 0 1 0 8 】

実施例 6 では、解析ボード 2 4 上に正常マイコン 4 5 を装着し、実施例 5 と同様に正常マイコントレースマップを収集する。収集した後、ワークステーションにおいて、正常マイコントレースマップと不具合マイコントレースマップとを比



較し、同値であれば0、異値であれば1となるトレース差分マップを生成する。

#### 【0109】

以上示したように、実施例6ではトレース差分マップにおいて「1」の箇所（差分発生箇所）は、正常マイコンと不具合マイコンの値が異なる箇所にある明確な不具合を示しているため、差異発生箇所に注目して解析作業を行うことによって解析精度が向上する。

#### 【0110】

##### （実施例7）

実施例7は、1. 不具合の再現性が悪い、という問題と、2. 真の不具合と検出された不具合との関連性の解析が困難である、という問題を解決する。実施例6において、差異箇所の明確化は可能になったが、不具合の要因になる事象が別の事象に従属している可能性があり、複数の発生例から解析が必要な場合が多い。本実施例では、複数の発生例から解析するデータ処理法を示す。

#### 【0111】

実施例7を、図28のトレース差分マップの比較説明図によって説明する。本実施例では、不具合マイコンを使用して、不具合発生例を6例発生させ、各々の発生例に関して実施例5と同様にトレース差分マップを求めた。各不具合例において、リセット～不具合検知までのタイミングは不定であるが、不具合発生を検知した端子トレースデータの差異点を一致させるように各マップをアライメントすることにより、不具合検知前のトレース状態を比較することが可能になる。図28中に、アライメントした6例のデータの平均値の分布を「トレース差分マップ間平均分布」、ばらつき（3σ）の分布を「トレース差分マップ間3σ分布」として示している。

#### 【0112】

実施例7では、得られた6例のトレースマップを、ワークステーションにおいて、端子トレースデータで最初に差異が発生した箇所を合わせるアライメント操作を行い、最も狭いマップに正規化する。次に、正規化されたマップに対し、マップ間の全トレース点の平均と3σ値のマップを、統計処理することにより得る。

## 【 0 1 1 3 】

以上示したように、実施例 7 では、平均のマップと  $3\sigma$  のマップとにより、確実に差異が発生する箇所、発生例により変動がある箇所等のグルーピングを行うことで、主事象、従属事象の判別が可能になり、従属事象が発生している場合であっても解析が可能になる。

## 【 0 1 1 4 】

## (実施例 8)

実施例 8 は、1. 不具合の再現性が悪い、という問題を解決する。実施例 5 によるトレースデータの取得において、リセットから不具合発生が長時間に及ぶ場合、トレースの取得において、長時間の再生を繰り返す必要があり、トレースを得るために長時間を必要とする。そこで実施例 8 では、任意のマイコンの状態を取得／レジューム（復活）させることにより、繰り返し時間を短縮することを目的とする。

## 【 0 1 1 5 】

実施例 8 において、不具合データ記録時の構成は実施例 1 と同様である。また、不具合データ再生時の動作も実施例 1 と同様である。図 29 に、実施例 8 における、復活データ取得時の説明図を示す。不具合データの再生後、トレース取得タイミング S より前のタイミングに復活タイミング R を設定する。

## 【 0 1 1 6 】

実施例 8 では、実施例 5 と同様の方法で復活タイミング R におけるレジスタ、RAM トレースを取得し、復活データ 89 として保存する。図 30 は、状態の復活とトレース取得の状態を示す図である。図 30 において、復活タイミング R における端子状態をマイコンに出力し、デバッグモニタを使用し、復活データの RAM 状態を順次マイコンに書き込む。次に、タイマ等の自前で変化するレジスタを残し、レジスタ状態を書き込む。最後に残ったレジスタを書き込み、保存していた状態が復活可能となる。復活した後、トレースデータ取得タイミング S に達すると、トレースデータを取得する。

## 【 0 1 1 7 】

以上示したように、実施例 8 では、状態を復活することにより再生時間を大幅

に短縮することができ、トレースデータの取得時間の短縮化が図れる。

【0118】

(実施例9)

実施例9は、1. 不具合の再現性が悪い、という問題と、2. 真の不具合と検出された不具合との関連性の解析が困難である、という問題を解決する。実施例5によって、不具合／正常マイコントレースデータの取得が可能になったが、端子、RAM、レジスタトレースデータではCPUが処理している命令は判明しない。実施例9では、トレースデータと実行中の命令との融合を目的とする。

【0119】

実施例9のシステムを、図31によって説明する。正常マイコン再生時に、正常マイコンの変わりにエミュレータ46を使用する。エミュレータ機能により、入出力端子の状態トレース及び命令トレースが取得可能である。

【0120】

実施例9では、エミュレータ46によるトレースデータ90、91を、ホストコンピュータ49により取得し、ワークステーションのストレージ88に転送する。トレースデータ151には、命令トレースデータとレジスタトレースデータが含まれる。

【0121】

既に取得済みのレジスタトレースデータとエミュレータ46によるレジスタトレースデータの状態を比較し、アライメントを行い、オフセットを取得する。このオフセット命令をトレースデータにも適用することにより、同期した、端子－レジスタ－RAM－命令、のトレースを得ることが可能になる。

【0122】

以上示したように、実施例9では、命令トレースデータを同期させることにより、より詳細にマイコンの動作を把握することが可能になり、解析精度が向上する。

【0123】

(実施例10)

実施例10は、1. 不具合の再現性が悪い、という問題を解決する。不具合の

解析を行う場合、実際の処理速度では速すぎて不具合を見落とすことが発生していた。実施例 1 0 はこの問題を解決する。

## 【 0 1 2 4 】

実施例 2 における記録に関して、図 3 2 に解析ボード構成図を示す。記録時に関しては、実速度 1 倍にて記録する。また、実施例 2 における再生に関して、図 3 3 に解析ボード構成図（再生時）を示す。

## 【 0 1 2 5 】

実施例 1 0 では、再生時に、クロック制御ボードをワークステーションにより操作し、クロック周波数を調整する。不具合を見落とす場合は、クロックスピードを低下させる。記録時は外部回路があるため実速度でしか動作しない場合があるが、再生時はマイコンに供給するクロックスピードを可変するだけであるため、マイコンが許容する範囲内で自由にクロック速度が可変可能である。

## 【 0 1 2 6 】

以上示したように、実施例 1 0 では、再生時のクロックスピードを可変することにより、記録時には発見できなかった不具合が発見しやすくなる。また、実速度では再生に長時間を要する場合には、クロック速度を上げることにより再生時間を短縮できる。

## 【 0 1 2 7 】

## （実施例 1 1）

実施例 1 1 は、1. 不具合の再現性が悪い、という問題と、3. 不具合の再発を防止するテスト方法の確立が困難である、という問題を解決する。従来は、マイコンの不具合を発見してから解析作業を行い、原因を究明した後に不具合が発生したマイコンを検査で失格とするテストプログラムを作成していた。しかしこの作成には非常に時間を要していた。実施例 1 1 は、短時間に不具合マイコンを失格とする検査方法を提供する。

## 【 0 1 2 8 】

実施例 1 1 による検査方法を、図 3 4 の解析ボード構成図に示す。実施例 1 1 における検査方法は、実施例 2 と同様に、不具合マイコン 2 2 の不具合発生の端子トレースデータを取得し、正常マイコンで再生させ、正常マイコンの端子トレ

ースデータを取得しておく。そして検査時には、正常マイコンの端子トレースデータを、評価マイコン 9 2 によって再生する。

#### 【 0 1 2 9 】

評価マイコン 9 2 が正常であった場合、マイコンの端子出力と端子トレースデータの差異は無いため、差異信号は” L ” のままで再生が終了する。評価マイコン 9 2 に不具合マイコン 2 2 と同等の不具合があった場合は、再生時にマイコンの出力端子と端子トレースデータとの差異が発生し、差異信号が” L ” から” H ” に変化し、異常が発生したことを検知できる。

#### 【 0 1 3 0 】

以上のように、実施例 1 1 の検査方法により、不具合の原因を解析できていない段階であっても、不具合の発生するマイコンを検査することが可能になる。また、応用として、図 3 5 に示すように、解析ボード 2 4 を複数用意し、各解析ボードの差異信号を多入力（マルチゲート）O R 9 4 に入力し、論理和の信号を差異信号としてワークステーションに接続することにより、複数のマイコンの検査を同時に行うことが可能になり、検査スループットの向上が期待できる。

#### 【 0 1 3 1 】

##### （実施例 1 2）

以上説明した実施例 1 ～ 1 1 に関しては、デジタル入力端子のみのマイコンに関して適応する。実施例 1 2 は、アナログ入力端子対応＝A D 変換付きのマイコンに適用する。以上示したように、実施例 1 2 では、記録時にはアナログ入力端子の印加電圧を記録する。そして再生時には、再生した電圧をアナログ入力端子に印加することにより、アナログ入力端子付きマイコンに対応する。以下に、実施例 1 2 の記録時、再生時の構成を説明する。

#### 【 0 1 3 2 】

記録時の、アナログ入力端子付きマイコンに対応したワークステーションに関し、図 3 6 に、アナログ端子対応ワークステーション構成図（記録時）を示す。実施例 2 のワークステーションに対して、下記の項目が追加される。

#### 【 0 1 3 3 】

アナログ I O ボード 1 0 3 は、アナログ対応記録ソフトウェア 9 8 の制御によ

って、アナログ入力ポート 1 0 4 に対して A D 変換を行い、変換結果をストレージ 9 7 上の記録アナログトレースデータ 1 0 0 に記録する。アナログ端子入力線はアナログ入力ポート 1 0 4 に接続されており、アナログ対応記録ソフトウェア 9 8 の制御によりアナログ入力ポート 1 0 4 が入力状態に設定され、A D 変換を行う。アナログ端子出力線はアナログ出力ポート 1 0 5 に接続されており、アナログ対応記録ソフトウェア 9 8 の制御によりアナログ出力ポート 1 0 5 が停止（ハイインピーダンス）状態に設定されている。記録アナログトレースデータ 1 0 0 は、アナログ入力ポート 1 0 4 に入力された電圧のトレースデータである。

【 0 1 3 4 】

次に、アナログ端子付きマイコンに対応した解析ボードに関して、図 3 7 のアナログ対応解析ボード構成図（記録時）によって説明する。実施例 1 2 では、実施例 2 の解析ボードに対して次の項目が付加される。アナログ入力端子 9 3 は、マイコンのアナログ入力端子である。アナログ端子入力線は、アナログ入力端子 9 3 の印加電圧を送出する。アナログ端子出力線は、ハイインピーダンス（停止）状態に設定されるため無効である。上記 2 線は、マイコンソケット 3 6 のアナログ入力端子 9 3 に接続されており、互いに導通している。

【 0 1 3 5 】

実施例 1 2 における再生時について、図 3 8 に、アナログ入力端子対応ワークステーション構成図（再生時）を示して説明する。

【 0 1 3 6 】

記録時のワークステーションに対して、次の項目が変更されている。出力アナログトレースデータ 1 0 6 は、アナログ対応再生ソフトウェア 9 8 の制御により、記録アナログトレースデータ 1 0 0 をコピーしたデータである。このデータに従って、アナログ入力端子 9 3 に印加する電圧を生成する。アナログ端子入力線は、アナログ対応再生ソフトウェアの制御により停止状態に設定されている。アナログ端子出力線は、アナログ対応記録ソフトウェア 9 8 の制御により出力状態に設定され、出力アナログトレースデータ 1 0 6 を順次 D A 変換した電圧を出力する。

【 0 1 3 7 】

次に、図 3 9 の、アナログ入力端子対応解析ボード構成図（再生時）によって説明する。記録時に対して次の項目が変更されている。アナログ端子入力線は、停止状態に設定されている。アナログ端子出力線は、出力状態に設定されており、アナログ入力端子に対し電圧を印加する。上記 2 線は、マイコンソケット 3 6 のアナログ入力端子 9 3 に接続されている。

## 【 0 1 3 8 】

実施例 1 2 において、記録時は、不具合マイコン 2 2 のアナログ入力端子に対象システムより電圧が印加される。この電圧を、アナログ入力端子→検査プローブ→マイコンソケット→アナログ入力端子線→アナログ入力ポート、の経路を経てアナログ I O ボード 1 0 3 に到達し、A D 変換された電圧データを、端子トレースデータ 9 9 と同様に、記録アナログトレースデータ 1 0 0 に保存する。

## 【 0 1 3 9 】

アナログ入力端子の再生においては、記録アナログトレースデータ 1 0 0 を出力アナログトレースデータ 1 0 6 にコピーした後、このデータをアナログ I O ボード 1 0 3 にロードし、端子トレースデータ 9 9 と同期して D A 変換を行い、アナログ電圧を、アナログ出力ポート→アナログ端子出力線→マイコンソケット→アナログ入力端子、を経由してアナログ入力端子 9 3 に印加する。この動作により、正常マイコン 4 5 のアナログ入力端子 9 3 には、記録時と同じ電圧が印加される。

## 【 0 1 4 0 】

以上示したように、実施例 1 2 では、アナログ入力端子をもつマイコンにおいても、記録、再生、解析が可能になる。

## 【 0 1 4 1 】

## （実施例 1 3）

実施例 1 2 において、アナログ入力端子に対応する方法を説明したが、実施例 1 3 では、アナログ入力端子＝D A 変換付きマイコンに適応することを目的とする。実施例 1 3 では、記録時には、アナログ出力端子 1 5 5 の印加電圧を記録する。そして再生時には、アナログ出力端子 1 5 5 より出力された電圧と記録時の電圧の比較を行う。

## 【 0 1 4 2 】

以下に、各々の状態に対応した構成を説明する。記録時は、実施例 1 2 と同様である。再生時は、図 4 0 にアナログ出力端子対応ワークステーション構成図（再生時）を示して説明する。記録時のワークステーションに対して、次の項目が変更されている。再生アナログトレースデータ 1 0 7 は、アナログ I O ボード 1 0 3 によりアナログ端子入力線の電圧を A D 変換したデータを保存する。

## 【 0 1 4 3 】

次に、図 4 1 にアナログ出力端子対応解析ボード構成図（再生時）を示して説明する。記録時に対して変更はない。記録時は、実施例 1 2 と同様である。再生時は、記録時と同様に、アナログ出力端子 1 5 5 の電圧を再生アナログトレースデータ 1 0 7 に記録する。再生完了後に、記録時に記録した記録アナログトレースデータ 1 0 0 と、再生時に記録した再生アナログトレースデータ 1 0 7 との比較を行う。

## 【 0 1 4 4 】

この比較を、図 4 2 のアナログ出力端子トレースデータ比較図で説明する。図 4 2 において、端子トレースデータ（記録時）、記録アナログトレースデータ、アナログトレースデータ差分、再生アナログトレースデータを示す。記録時に取得した不具合マイコン 2 2 の記録アナログトレースデータを比較すると、差異点 E においてトレース電圧が異なっており、この時点でアナログ出力電圧が異なるという不具合が検知される。

## 【 0 1 4 5 】

以上示したように、実施例 1 3 では、アナログ出力端子 1 5 5 を持つマイコンにおいて、記録、再生、解析等の動作が可能となり、アナログ出力電圧が異なる不具合に対しても検知可能となる。

## 【 0 1 4 6 】

## （実施例 1 4）

実施例 1 4 は、1. 不具合の再現性が悪い、という問題を解決する。実施例 1 2 において、アナログ入力端子をもつマイコンにおける記録、再生、解析動作が可能になったが、A D 変換においては、マイコン毎の A D 変換個体差が存在し、



同じ端子電圧を印加した場合であっても変換値が異なる事象が発生し、再生、解析動作が正常に行われず。実施例 1 4 では、このマイコンの個体差を吸収する方法を提供する。

#### 【0 1 4 7】

実施例 1 4 は、基本的に比較するマイコン相互のアナログ差分特性を収集しておき、トレース使用前に補正をかける方法である。アナログ入力端子の A D 変換特性を評価するために、A D 変換器の変換値＝アナログ入力レジスタを読む必要があるため、デバッグモニタを搭載したマイコンを使用する。図 4 3 に、デバッグモニタを使用したマイコンの内部構造を示す。デバッグモニタを搭載したマイコンに関して説明した実施例 4 と異なる点は、アナログ入力端子 1 1 9 と、アナログ入力端子電圧を A D 変換した値を保持するアナログ入力レジスタ 1 1 4 がある点である。

#### 【0 1 4 8】

実施例 1 4 では、最初に、正常マイコンと不具合マイコンの A D 差分特性を取得する。解析ボード 2 4 上のマイコンソケット 3 6 に正常マイコンを搭載し、ワークステーションにおいて、アナログ入力補正ソフトウェア 1 2 3 を動作させ、アナログ I O ボードに測定範囲の初期値（本実施例では、0 H）を設定（アナログ入力レジスタ読み出しコマンドをシリアル入力し、読み出し値のレスポンスを得る）し、対応するアナログ電圧を出力し、マイコンのアナログ入力端子に印加する。

#### 【0 1 4 9】

次に、デバッグモニタの機能を利用して、マイコン上で A D 変換を行い、A D 変換値＝アナログ入力レジスタを取得する。次に、アナログ I O ボードの設定値をインクリメント（0 1 H）し、出力電圧を変化させ、同様にアナログ入力レジスタを取得する。この動作を上限（本実施例では、F F F F H）まで繰り返し、アナログ I O ボードの出力設定値と、アナログ入力レジスタ値との相関を取得する。図 4 4 は、アナログ I O ボードとアナログ入力レジスタ値との相関図である。アナログ I O ボードの出力設定値を、図 4 4 の（A）に示す。

#### 【0 1 5 0】

次に、搭載するマイコンを、不具合マイコンに交換して同様に相関を取得する。アナログ入力レジスタ値を、図 4 4 の (C) に示す。これら両相関より、相関特性の差分 (図 4 4 の (B)) が得られる。この差分より、アナログ変換特性を補正することが可能になる (図 4 5)。

#### 【 0 1 5 1 】

次に、実施例 1 2 と同様に、不具合マイコンに対する記録を行い、記録アナログトレースデータを取得する。実施例 1 2 においては、記録アナログトレースデータを無加工のまま出力アナログトレースデータに複製したが、本実施例では、記録アナログトレースデータに対し、差分特性による補正を行い、出力アナログトレースデータを生成する。この補正により、正常マイコンと不具合マイコンの A/D 変換後のアナログ入力レジスタ値は一致するはずである。そして再生時には、生成した出力アナログトレースデータを使用する。

#### 【 0 1 5 2 】

以上示したように、実施例 1 4 では、アナログ入力特性の異なる不具合マイコンと正常マイコンにおいても、差分による補正を行うことによって同等の再生状態を得ることができる。

#### 【 0 1 5 3 】

##### (実施例 1 5)

実施例 1 5 は、1. 不具合の再現性が悪い、という問題を解決する。実施例 1 3 において、アナログ出力端子 1 5 5 を持つマイコンにおける記録、再生、解析動作が可能になったが、D/A 変換においては、マイコン毎の D/A 変換個体差が存在し、同じ値を設定した場合であっても、出力電圧が異なる事象が発生し、再生、解析動作が正常に行われない場合があった。実施例 1 5 では、この個体差を吸収する方法を提供する。

#### 【 0 1 5 4 】

実施例 1 5 では、比較するマイコン相互のアナログ差分特性を収集しておき、トレース比較において補正をする。アナログ出力端子の D/A 変換特性を評価するために、D/A 変換器の設定値 = アナログ出力レジスタを書き込む必要があるため、デバッグモニタを搭載したマイコンを使用する。図 4 6 に、デバッグモニタを

使用したマイコンの内部構造を示す。

【 0 1 5 5 】

デバッグモニタを搭載したマイコンに関して説明した実施例 4 と異なる点は、アナログ出力端子 1 2 9 と、アナログ出力電圧の設定値を保持するアナログ出力レジスタ 1 5 2 がある点である。

【 0 1 5 6 】

実施例 1 5 では、最初に、正常マイコンと不具合マイコンの D A 差分特性を取得する。まず、解析ボード上のマイコンソケットに正常マイコンを搭載し、ワークステーションにおいて、アナログ出力補正ソフトウェア 1 2 4 を動作させ、デバッグモニタ 1 1 1 を操作し、アナログ出力レジスタ 1 5 2 に設定範囲の初期値（本実施例では、0 H）を設定し（アナログ出力レジスタ書き込みコマンドをシリアルに入力し、書き込み完了レスポンスを得る。）、D A 変換動作をさせ、アナログ出力端子に D A 変換電圧を出力させ、この電圧をアナログ I O ボード 1 0 3 で A D 変換し、A D 変換値を読みとる。

【 0 1 5 7 】

次に、アナログ出力レジスタ値をインクリメント（0 1 H）し、出力電圧を変化させ、同様にアナログ I O ボード 1 0 3 の A D 変換値を読みとる。この動作を上限（本実施例では、F F H）まで繰り返し、アナログ出力レジスタ 1 5 2 の設定値とアナログ I O ボードの A D 変換値の相関を取得する。図 4 7 は、アナログ出力設定値とアナログ I O ボード入力値との相関図である。図 4 7 中に、アナログ出力設定値を（A）に示す。

【 0 1 5 8 】

次に、搭載するマイコンを不具合マイコンに交換して同様に相関を取得する。アナログ I O ボード入力値を、図 4 7 の（C）に示す。これら両者の相関より、相関特性の差分ず 4 7 の（B）が得られ、この差分によりアナログ変換特性を補正することが可能になる。

【 0 1 5 9 】

次に、実施例 1 3 と同様に不具合マイコンに対する記録を行い、差分特性による補正を行い、比較アナログトレースデータを生成する。この補正により、比較

アナログトレースデータは不具合マイコンのアナログ出力値と一致する。再生時には、実施例 1 3 と同様に再生を行い、再生アナログトレースデータを取得する。次に、実施例 1 3 と同様に、比較アナログトレースデータと再生アナログトレースデータの比較を行う（図 4 8）。

## 【 0 1 6 0 】

以上示したように、実施例 1 5 では、アナログ出力特性の異なる不具合マイコンと正常マイコンにおいても、差分による補正を行うことによって個体差に依存しない比較を行うことができる。

## 【 0 1 6 1 】

## （実施例 1 6）

実施例 1 6 は、1. 不具合の再現性が悪い、という問題と、2. 真の不具合と検出された不具合との関連性の解析が困難である、という問題を解決する。実施例 4 においては、デバッグモニタを使用してマイコンの内部情報を得たが、デバッグモニタはソフトウェアであるためアプリケーションソフトウェアとのリソース競合が発生し、共存が難しい欠点があった。実施例 1 6 は、オンチップデバッグの搭載により、上記の問題を回避することを目的とする。

## 【 0 1 6 2 】

図 4 9 に、オンチップデバッグ 1 3 4 を搭載したマイコンの内部構成図を示す。実施例 1 6 のオンチップデバッグ 1 3 4 は、デバッグ専用のインターフェイス 1 3 7 をもち、このデバッグインターフェイス 1 3 7 を起動することによりアプリケーションソフトウェア 1 2 7 を停止し、RAM 1 2 9、レジスタ 1 3 0、1 3 2 等のマイコン 1 2 6 のリソースが操作可能な機能をもたせる。

## 【 0 1 6 3 】

図 5 0 に、このオンチップデバッグに対応したワークステーションの構成図（解析時）を示す。実施例 4 からは次の点が変更されている。デバッグインターフェイスボード 1 5 0 は、オンチップデバッグ 1 3 4 に対応した双方向インターフェイスである。デバッグインターフェイス信号線は、マイコン 1 2 6 のデバッグインターフェイス 1 3 7 とワークステーション 1 4 0 のデバッグインターフェイスボード 1 5 0 とを接続する信号線である。

【 0 1 6 4 】

実施例 1 6 では実施例 4 と同様に、解析時にマイコン内部の情報を取得し、書き込み時にデバッグモニタの変わりにオンチップデバッガを操作する。

【 0 1 6 5 】

以上示したように、実施例 1 6 では、モニタデバッガはプログラム中に搭載する必要があるため、ソフトウェア中にデバッグモニタを搭載しないマイコンに関しては本評価システムを使用できない。また、個々のアプリケーションにおいて、デバッグモニタに使用するリソースが変わり、インターフェイスが変わる可能性があったが、オンチップデバッガを搭載したマイコンを使用する場合は、該当マイコン全てに適用可能で、また、同じデバッグインターフェイスをもつため、デバッグインターフェイスボードを共有でき、適用範囲が広がる。

【 0 1 6 6 】

(実施例 1 7)

実施例 1 7 は、4. 解析システムの構築が困難である、という問題を解決する。以上の実施例 1 ～ 1 6 では、本発明によるシステムを構成するためにはワークステーションで使用する以下の多数のソフトウェアを開発する必要がある。例えば、

- ・記録ソフトウェア（モジュールあり、デジタル）
- ・記録ソフトウェア（モジュールあり、デジタル、アナログ）
- ・記録ソフトウェア（モジュールなし、デジタル）
- ・記録ソフトウェア（モジュールなし、デジタル、アナログ）
- ・再生ソフトウェア（同上）
- ・解析ソフトウェア（同上＋モニタデバッガ対応）
- ・解析ソフトウェア（同上＋オンチップデバッガ対応）
- ・自動トレースソフトウェア（同上）
- ・復活ソフトウェア
- ・ . . . . . などである。

【 0 1 6 7 】

各々に関して、対象マイコン毎にアナログの有無、デバッグモニタの有無、イ

ンターフェイスの種類等の複雑なパラメータがあり、全てのソフトウェアを開発するためには多大な開発工程が必要となる。そこで、実施例 17 では、これらのソフトウェアの開発効率を上げるソフトウェア構造を提供することを目的とする。

#### 【0168】

図 5 1 に、ホストソフトウェアの構造を示す。デバッグモニタ／オンチップデバッグの違いにより、デバッグインターフェイスが、シリアル I O / デバッグインターフェイスの 2 種類が存在する。図 5 1 中、ソフトウェアの囲い内は、ワークステーションのソフトウェアの内部構造に関する項目である。

#### 【0169】

ソフトウェアは 3 階層に構造化される。最下層がドライバレイアであり、この階層において、搭載するボードに対応するドライバを提供する。クロック制御ボードに関しては、クロック制御ボードドライバが対応する。

#### 【0170】

ドライバレイアの上位には、機能モジュールレイアが存在し、その間には、ドライバ A P I (アプリケーションインターフェイス) が規定されており、このインターフェイスを介して、プロセス間通信を行う。

#### 【0171】

ドライバレイアの目的は、異なったボードが搭載された場合であってもドライバ内部でボードの差異を吸収し、同一のドライバ A P I を提供することである。例として、デバッグインターフェイスとして実施例 4 で使用したシリアル I O が用いられているが、これを実施例 16 で使用したオンチップデバッグに変える場合、インターフェイスボードとドライバの両方を交換することにより、ハードウェアの差をドライバが吸収し、デバッグに関するドライバ A P I は変えずに済むため、機能モジュールレイアのソフトウェアモジュールは変更する必要がない。

#### 【0172】

機能モジュールレイアの上位にはアプリケーションレイアが存在し、その間には、モジュール A P I が規定されており、このインターフェイスを介して、プロセス間通信を行う。モジュールレイアの目的は、デジタル端子記録処理、デジタ

ル端子再生処理等のソフトウェアを作成する上で必要となる機能単位での処理モジュールを展開することにある。

【 0 1 7 3 】

機能モジュールは、ハードウェアのアクセスに関しては、ドライバ A P I を介してドライバとプロセス間通信を行い、アプリケーションレイアに対しては、モジュール A P I を提供する。

【 0 1 7 4 】

アプリケーションレイアは、上に列挙した記録ソフトウェア、再生ソフトウェア等のアプリケーションを提供することを目的とし、モジュール A P I を介して機能モジュールをアクセスすることにより機能の実装を簡易化する。

【 0 1 7 5 】

実施例 1 7 において実際のソフトウェア開発は、ドライバ A P I、モジュール A P I の規定を明確にした上で必要とするハードウェア、個別機能、アプリケーションに関して、個別にドライバ、モジュール、アプリケーションを開発する。

【 0 1 7 6 】

以上説明したように、実施例 1 7 によるソフト階層構造を実現することにより、各ドライバ、モジュール、アプリケーションの開発が独立に行え、共通したりソースが使用可能であるため、開発効率がよく、柔軟性に富む評価システムの開発が可能になる。

【 0 1 7 7 】

(実施例 1 8)

実施例 1 8 は、2. 真の不具合と検出された不具合との関連性の解析が困難である、という問題を解決する。対象システムにおいて、マイコン以外の不具合について、再現性が乏しく実速度では減少を補足しきれない等の理由により解析が困難である場合があった。そこで、本発明によるシステムをマイコンを搭載する対象システムに適用すれば、システム全体の不良解析が可能になる。

【 0 1 7 8 】

図 5 2 に、対象システムへの応用例を示す。解析ボードを検査プローブを介してマイコンを除去したシステムに接続する。

【 0 1 7 9 】

あらかじめ取得しておいた不具合の発生を記録した端子トレースを再生することにより、システムの不具合が何度でも再生することができる。また、クロック速度を低下することにより、マイコンから外部に送出する信号速度が低下し、システムの不具合減少を補足しやすくなる。

【 0 1 8 0 】

以上示したように、実施例 1 8 により、マイコン搭載システムにおいて、解析しやすい速度で確実に不具合が再現可能である。このため、システムとしての不具合の解析効率が向上する。

【 0 1 8 1 】

以上、添付図面を参照しながら本発明のロジック L S I の不良解析システム、及び不良解析方法の好適な実施例について説明したが、本発明はこれらの例に限定されない。いわゆる当業者であれば、特許請求の範囲に記載された技術的思想の範疇内において各種の変更例または修正例に想到し得ることは明らかであり、それらについても当然に本発明の技術的範囲に属するものと了解される。

【 0 1 8 2 】

【発明の効果】

以上示したように、本発明によれば、不具合の再現性が良い、真の不具合と検出された不具合との関連性の解析が容易である、不具合の再発を防止するテスト方法の確立が容易である、解析システムの構築が容易である、ロジック L S I の不良解析システム、及び不良解析方法が提供できた。

【図面の簡単な説明】

【図 1】

図 1 は、実施例 1 における、不良解析システムの構成を示す概念図である。

【図 2】

図 2 は、実施例 1 における、記録時のワークステーションの構成を示す概念図である。

【図 3】

図 3 は、実施例 1 における、解析ボードの構成を示す概念図である。



【図 4】

図 4 は、実施例 1 における、マイコンインターフェイスモジュールの内部構成を示す概念図である。

【図 5】

図 5 は、実施例 1 における、端子トレースデータの記録図である。

【図 6】

図 6 は、実施例 1 における、再生時のワークステーションの構成を示す概念図である。

【図 7】

図 7 は、実施例 1 における、再生時の解析ボードの構成図である。

【図 8】

図 8 は、実施例 1 における、再生時のマイコンインターフェイスモジュールの構成を示す概念図である。

【図 9】

図 9 は、実施例 1 における、正常マイコン出力と不具合マイコン出力の端子トレースデータの記録図である。

【図 10】

図 10 は、実施例 1 における、記録ソフトウェアの動作を示すフローチャートである。

【図 11】

図 11 は、実施例 1 における、再生ソフトウェアの動作を示すフローチャートである。

【図 12】

図 12 は、実施例 2 における、記録時のワークステーションの構成を示す構成図である。

【図 13】

図 13 は、実施例 2 における、記録時の解析ボードの構成を示す概念図である。

【図 14】

図 1 4 は、実施例 2 における、再生時のワークステーションの構成を示す概念図である。

【図 1 5】

図 1 5 は、実施例 2 における、再生時の解析ボードの構成を示す概念図である。

【図 1 6】

図 1 6 は、実施例 2 における、端子トレースデータ比較図である。

【図 1 7】

図 1 7 は、実施例 3 における、不具合データ再生時のシステム構成図である。

【図 1 8】

図 1 8 は、実施例 4 における、マイコンの内部構成を示す概念図である

【図 1 9】

図 1 9 は、実施例 4 における、解析時のマイコンの内部構成を示す概念図である。

【図 2 0】

図 2 0 は、実施例 4 における、不具合データ解析時の解析ボードの構成を示す概念図である。

【図 2 1】

図 2 1 は、実施例 4 における、解析時のワークステーションの構成を示す概念図である。

【図 2 2】

図 2 2 は、実施例 4 における、解析ソフトウェアの動作を示すフローチャートである。

【図 2 3】

図 2 3 は、実施例 5 における、不具合データ解析時のシステムの構成を示す概念図である。

【図 2 4】

図 2 4 は、実施例 5 における、不具合データ解析時の動作を示す、自動トレースソフトウェアのフローチャートである。

【図 2 5】

図 2 5 は、実施例 5 における、状態取得中の端子トレースデータ、レジスタトレースデータ、RAMトレースデータの相関図である。

【図 2 6】

図 2 6 は、実施例 5 における、状態取得完了後の端子トレースデータ、レジスタトレースデータ、RAMトレースデータの相関図である。

【図 2 7】

図 2 7 は、実施例 6 における、トレース差分マップである。

【図 2 8】

図 2 8 は、実施例 7 における、トレース差分マップである。

【図 2 9】

図 2 9 は、実施例 8 における、復活データ取得の状態を示すマップである。

【図 3 0】

図 3 0 は、状態の復活とトレース取得の状態を示すマップである。

【図 3 1】

図 3 1 は、実施例 9 における、データ取得の状態を示すマップである。

【図 3 2】

図 3 2 は、実施例 1 0 における、記録時の解析ボードの構成を示す概念図である。

【図 3 3】

図 3 3 は、実施例 1 0 における、再生時の解析ボード構成を示す概念図である。

【図 3 4】

図 3 4 は、実施例 1 1 における、検査時の解析ボードの構成を示す概念図である。

【図 3 5】

図 3 5 は、実施例 1 1 における、複数の解析ボードの構成を示す概念図である。

【図 3 6】

図 3 6 は、実施例 1 2 における、記録時のアナログ端子対応ワークステーションの構成を示す概念図である。

【図 3 7】

図 3 7 は、実施例 1 2 における、記録時のアナログ対応解析ボードの構成を示す概念図である。

【図 3 8】

図 3 8 は、実施例 1 2 における、再生時のアナログ入力端子対応ワークステーションの構成を示す概念図である。

【図 3 9】

図 3 9 は、実施例 1 2 における、再生時のアナログ入力端子対応解析ボードの構成を示す概念図である。

【図 4 0】

図 4 0 は、実施例 1 3 における、再生時のアナログ出力端子対応ワークステーションの構成を示す概念図である。

【図 4 1】

図 4 1 は、実施例 1 3 における、再生時のアナログ出力端子対応解析ボードの構成を示す概念図である。

【図 4 2】

図 4 2 は、実施例 1 3 における、アナログ出力端子トレースデータの比較図である。

【図 4 3】

図 4 3 は、実施例 1 4 における、デバッグモニタを使用したマイコンの内部構造を示す概念図である。

【図 4 4】

図 4 4 は、実施例 1 4 における、アナログ I O ボードとアナログ入力レジスタ値との相関図である。

【図 4 5】

図 4 5 は、実施例 1 4 における、相関特性の差分から、アナログ変換特性を補正する動作を説明する概念図である。

【図 4 6】

図 4 6 は、実施例 1 5 における、デバッグモニタを使用したマイコンの内部構造を示す概念図である。

【図 4 7】

図 4 7 は、実施例 1 5 における、アナログ出力設定値とアナログ I O ボード入力値の相関図である。

【図 4 8】

図 4 8 は、実施例 1 5 における、比較アナログトレースデータと再生アナログトレースデータの比較を行う動作を説明する概念図である。

【図 4 9】

図 4 9 は、実施例 1 6 における、オンチップデバッグを搭載したマイコンの内部構成を示す概念図である。

【図 5 0】

図 5 0 は、実施例 1 6 における、解析時のオンチップデバッグに対応したワークステーションの構成を示す概念図である。

【図 5 1】

図 5 1 は、実施例 1 7 における、ホストソフトウェアの構造を示す概念図である。

【図 5 2】

図 5 2 は、実施例 1 7 における、対象システムへの応用例を示す概念図である。

【図 5 3】

図 5 3 は、従来の不良解析の対象システムに、不具合マイコンが組み込まれている状態を示す概念図である。

【図 5 4】

図 5 4 は、正常マイコンの内部構成を示す概念図である。

【図 5 5】

図 5 5 は、不具合マイコンの内部構成を示す概念図である。

【符号の説明】

- 1 対象システム
- 2 表示回路
- 3 入出力回路
- 4 内部回路
- 5 不具合マイコン
- 6 インターフェイス回路
- 7 マイコン
- 8 ROM
- 9 アプリケーションソフトウェア
- 1 0 CPU
- 1 1 RAM
- 1 2 インターフェイスレジスタ
- 1 3 ペリフェラルレジスタ
- 1 4 入出力インターフェイス
- 1 5 ペリフェラル
- 1 6 入力端子
- 1 7 出力端子
- 2 0 制御ブロック
- 2 1 対象システム
- 2 2 不具合マイコン
- 2 3 検査プローブ
- 2 4 解析ボード
- 2 5 接続ハーネス
- 2 6 ワークステーション
- 2 7 CPU
- 2 8 ストレージ
- 2 9 記録ソフトウェア
- 2 9' マイコンインターフェイスモジュール無し対応の記録ソフトウェア
- 3 0 再生ソフトウェア

- 3 1 端子トレースデータ（記録時）
- 3 1' 端子トレースデータ（再生時）
- 3 2 クロック制御ボード
- 3 3 端子デジタル I O ボード
- 3 4 制御デジタル I O ボード
- 3 5 マイコンインターフェイスモジュール
- 3 6 マイコンソケット
- 3 7 クロックブロック
- 3 8 端子制御ブロック
- 3 9 端子信号差異検出ブロック
- 4 0 端子切断制御部
- 4 1 制御信号デコーダ
- 4 2 クロック方向制御部
- 4 3 クロック入力バッファ
- 4 4 クロック出力バッファ
- 4 5 正常マイコン
- 4 6 エミュレータ
- 4 7 エミュレータプローブ
- 4 8 エミュレータケーブル
- 4 9 ホストコンピュータ
- 5 0 マイコン
- 5 1 ROM
- 5 2 アプリケーションソフトウェア
- 5 3 デバッグモニタソフトウェア
- 5 4 CPU
- 5 5 RAM
- 5 6 インターフェースレジスタ
- 5 7 シリアルインターフェースレジスタ
- 5 8 ペリフェラルレジスタ

- 5 9     インターフェイス
- 6 0     シリアルインターフェイス
- 6 1     ペリフェラル
- 6 2     入力端子
- 6 3     出力端子
- 6 4     シリアルインターフェイス入力端子
- 6 5     シリアル出力端子
- 6 6, 7 7, 9 5     ワークステーション
- 6 7, 7 8, 9 6     C P U
- 6 8, 7 9, 8 8, 9 7     ストレージ
- 6 9     解析ソフトウェア
- 7 0     端子トレースデータ
- 7 1     レジスタ状態データ
- 7 2     R A M 状態データ
- 7 3, 8 4     クロック制御ボード
- 7 4, 8 5     端子デジタル I O ボード
- 7 5, 8 6     制御デジタル I O ボード
- 7 6, 8 7     シリアル I O ボード
- 8 0     自動トレースソフトウェア
- 8 1     端子トレースデータ
- 8 2     レジストトレースデータ
- 8 3     R A M トレースデータ
- 8 9     復活データ
- 9 0     レジスタトレースデータ
- 9 1     R A M トレースデータ
- 9 2     評価マイコン
- 9 3     アナログ入力端子
- 9 4     マルチゲート O R
- 9 8     アナログ対応記録ソフトウェア

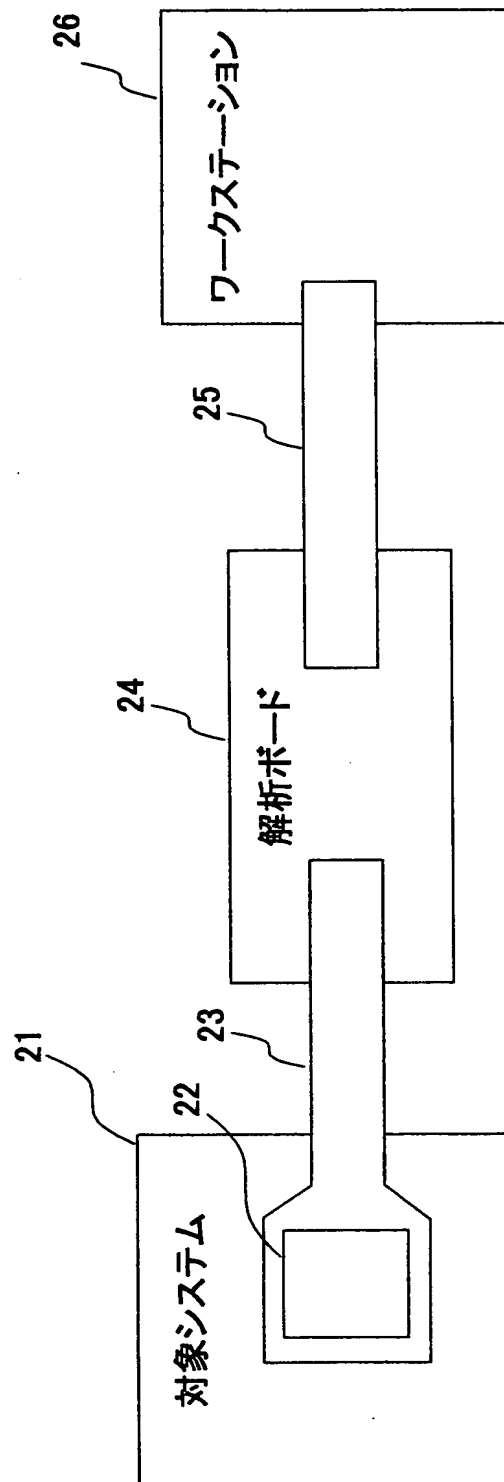


9 9	端子トレースデータ
1 0 0	記録アナログトレースデータ
1 0 1	クロック制御ボード
1 0 2	端子デジタル I O ボード
1 0 3	アナログ I O ボード
1 0 4	アナログ入力ポート
1 0 5	アナログ出力ポート
1 0 6	出力アナログトレースデータ
1 0 7	再生アナログトレースデータ
1 0 8, 1 2 6	マイコン
1 0 9	ROM
1 1 0	アプリケーションソフトウェア
1 1 1	デバッグモニタ
1 1 2, 1 2 8	CPU
1 1 3, 1 2 9	RAM
1 1 4	アナログ入力レジスタ
1 1 5	シリアルインターフェースレジスタ
1 1 6, 1 3 2	ペリフェラルレジスタ
1 1 7, 1 3 1	インターフェース
1 1 8, 1 3 3	ペリフェラル
1 1 9	アナログ入力端子
1 2 0	シリアル入力端子
1 2 1	シリアル出力端子
1 2 2	ストレージ
1 2 3	アナログ入力補正ソフトウェア
1 2 4	アナログ出力補正ソフトウェア
1 2 5	比較アナログトレースデータ
1 2 7	アプリケーションソフトウェア
1 2 9	アナログ出力端子

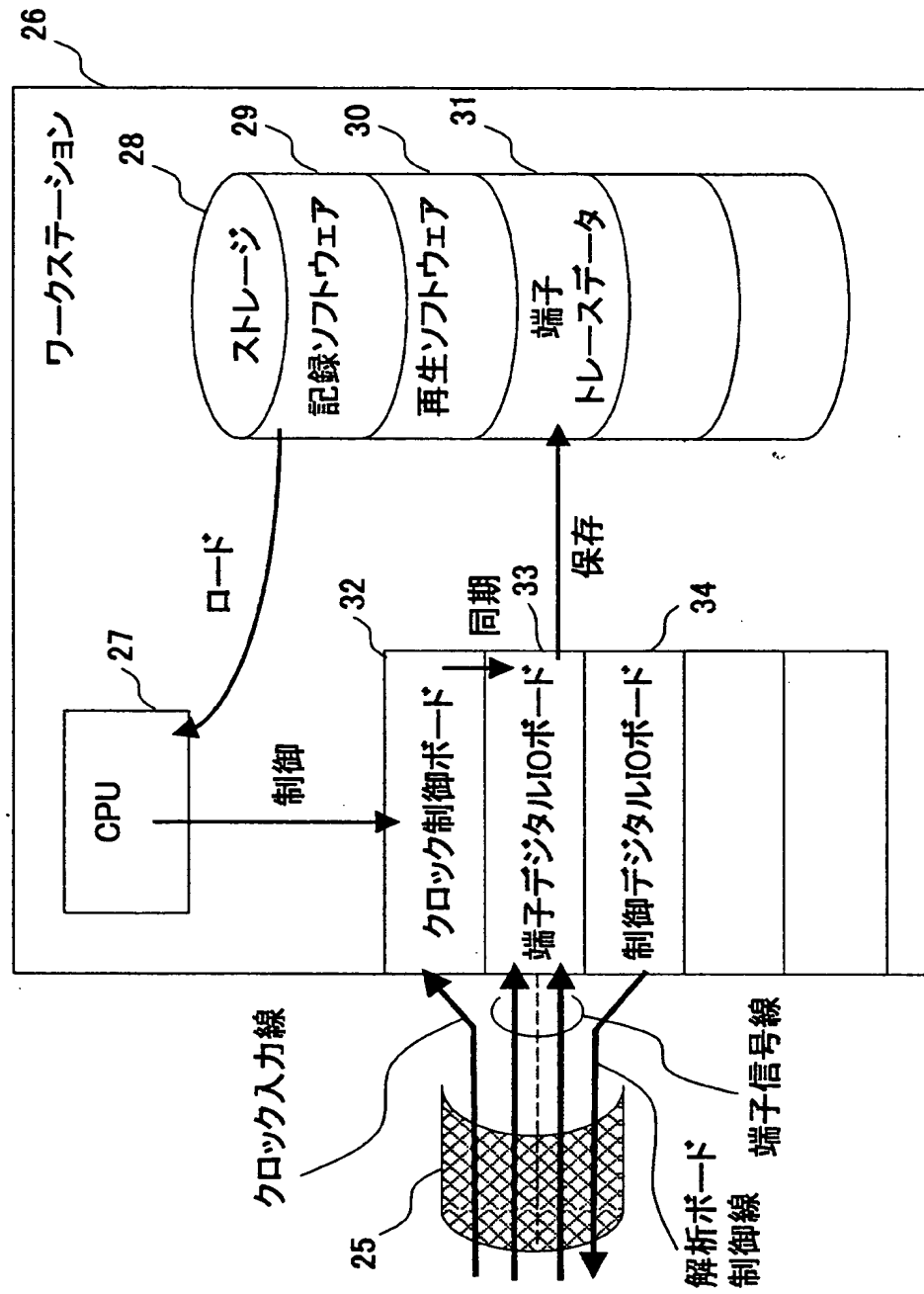
- 1 3 0     インターフェースレジスタ
- 1 3 4     オンチップデバッガ
- 1 3 5     入力端子
- 1 3 6     出力端子
- 1 3 7     デバッガインターフェイス
- 1 4 0     ワークステーション
- 1 5 0     デバッグインターフェイスボード
- 1 5 1     トレースデータ
- 1 5 2     アナログ出力レジスタ

【書類名】 図面

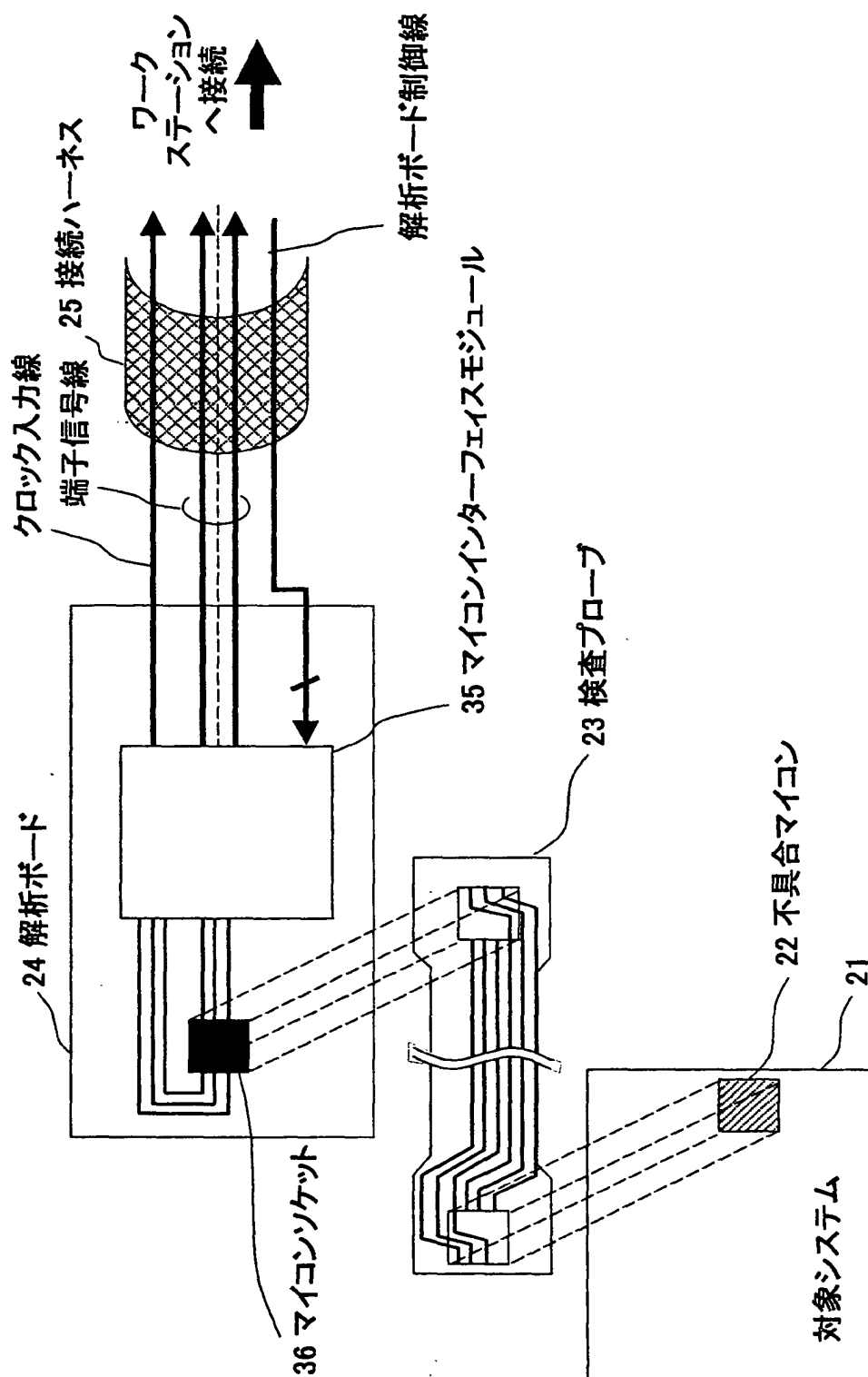
【図 1】



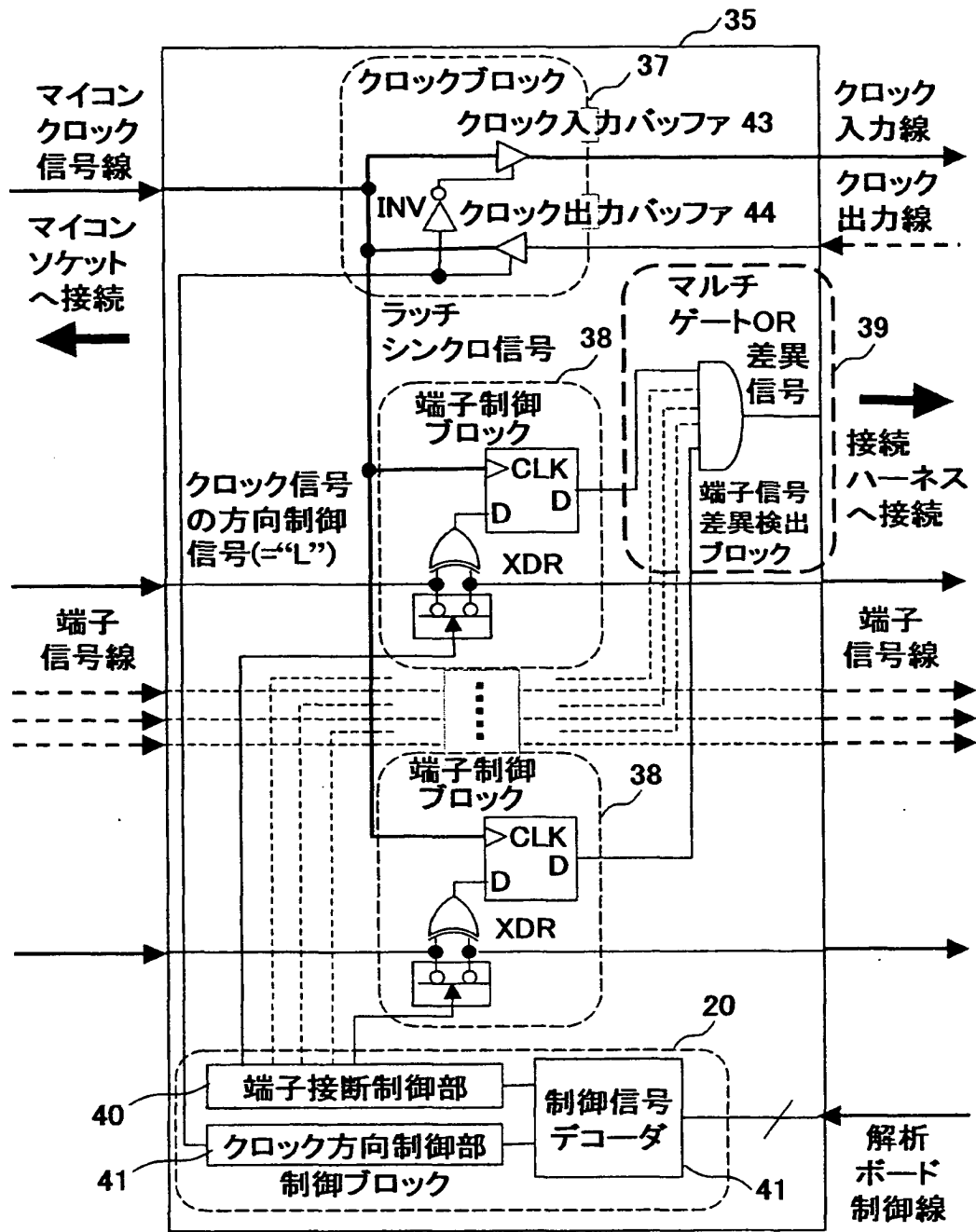
【図2】



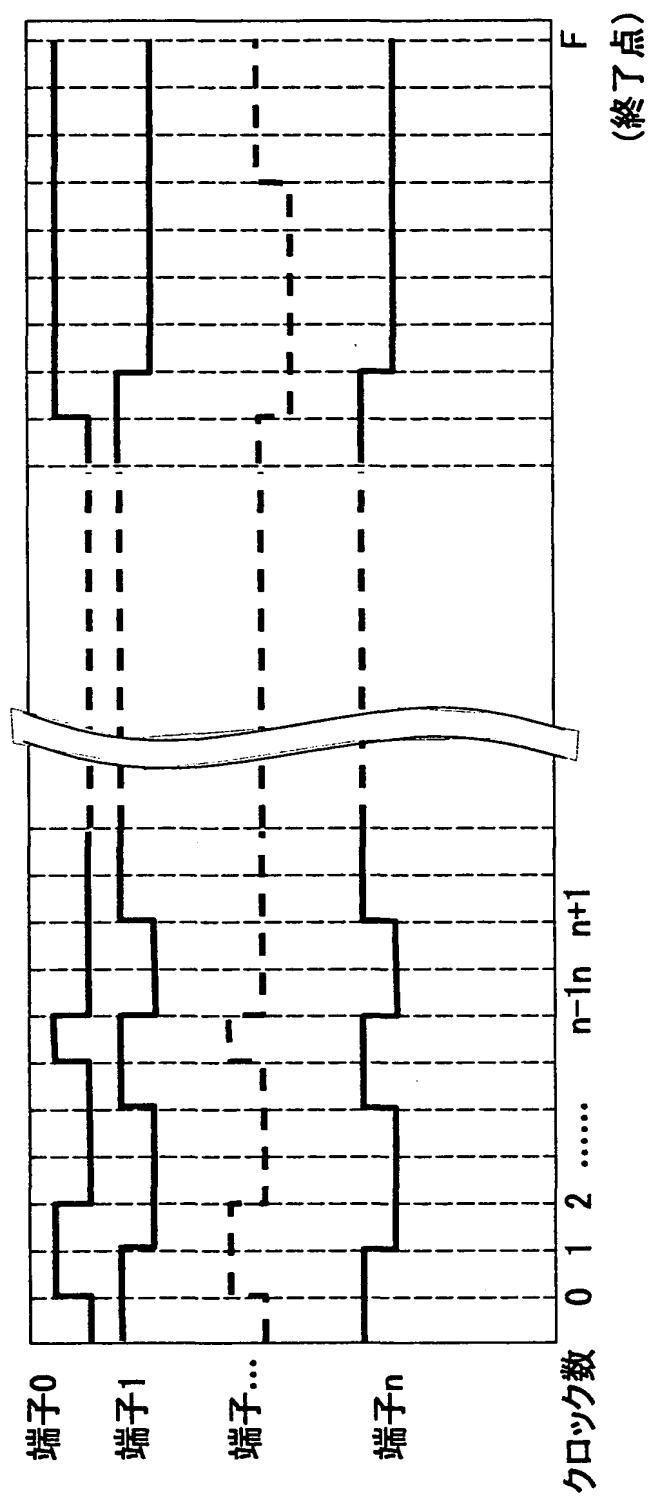
【図3】



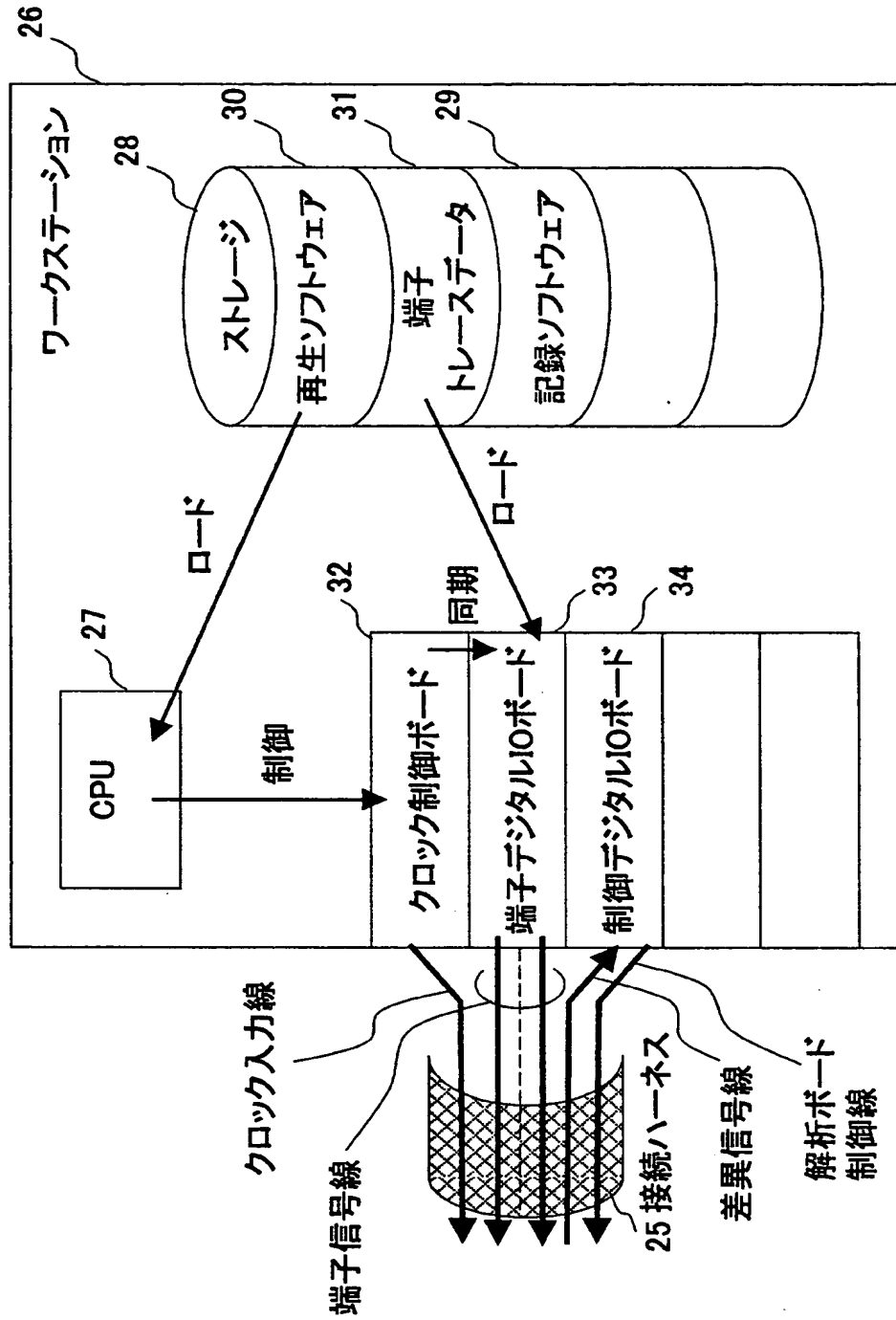
【図 4】



【図 5】

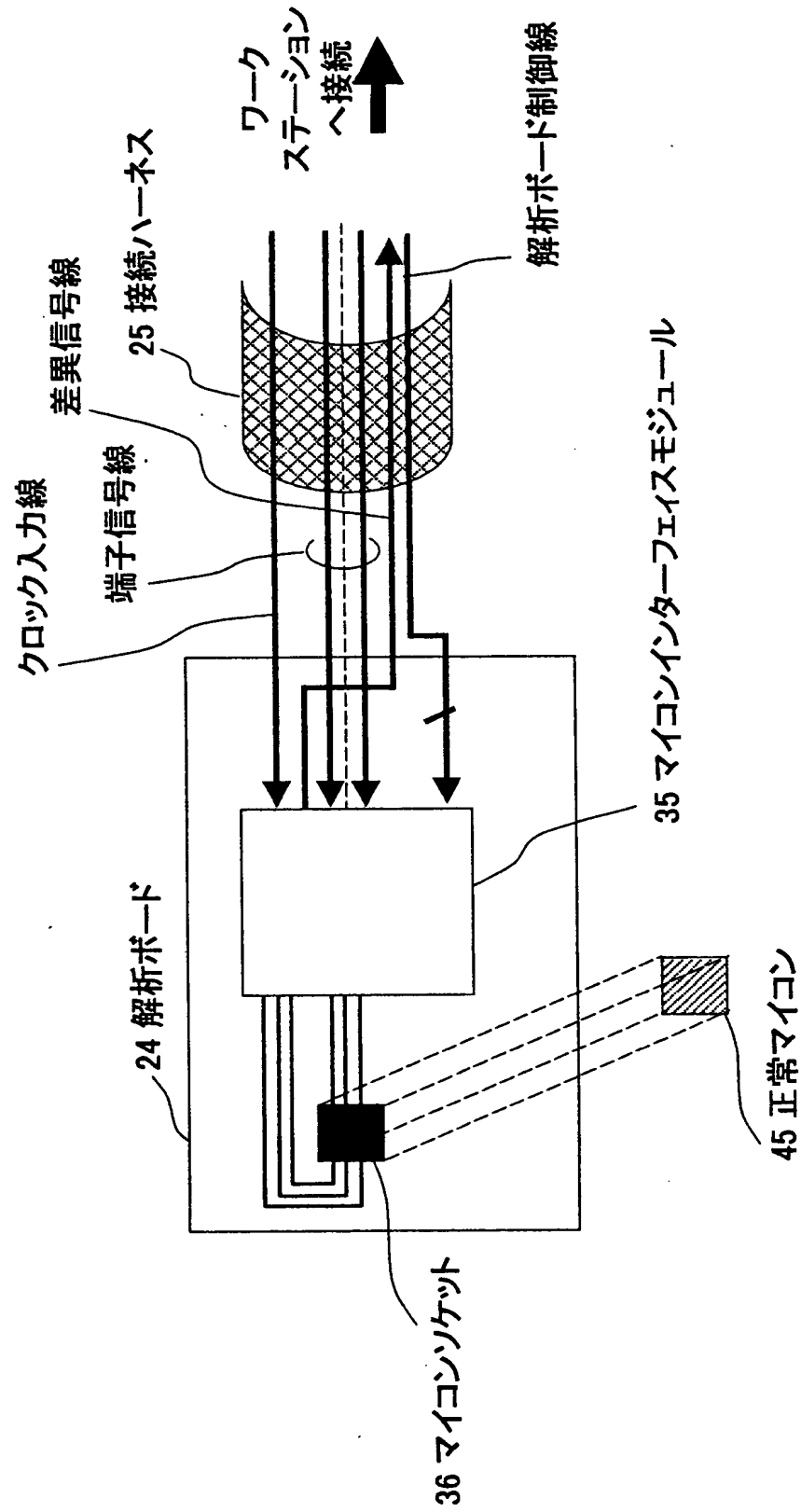


【図 6】

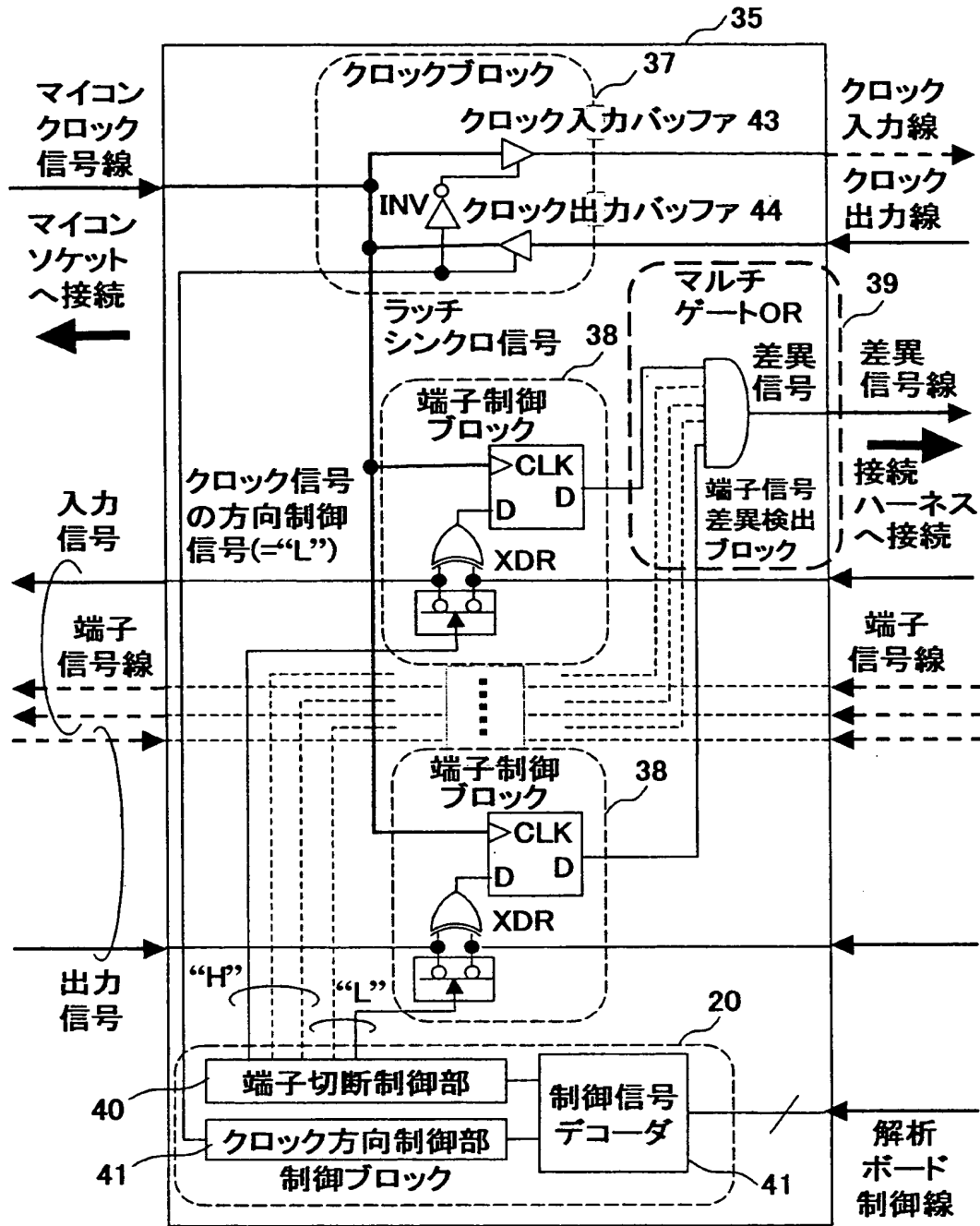


【図 7】

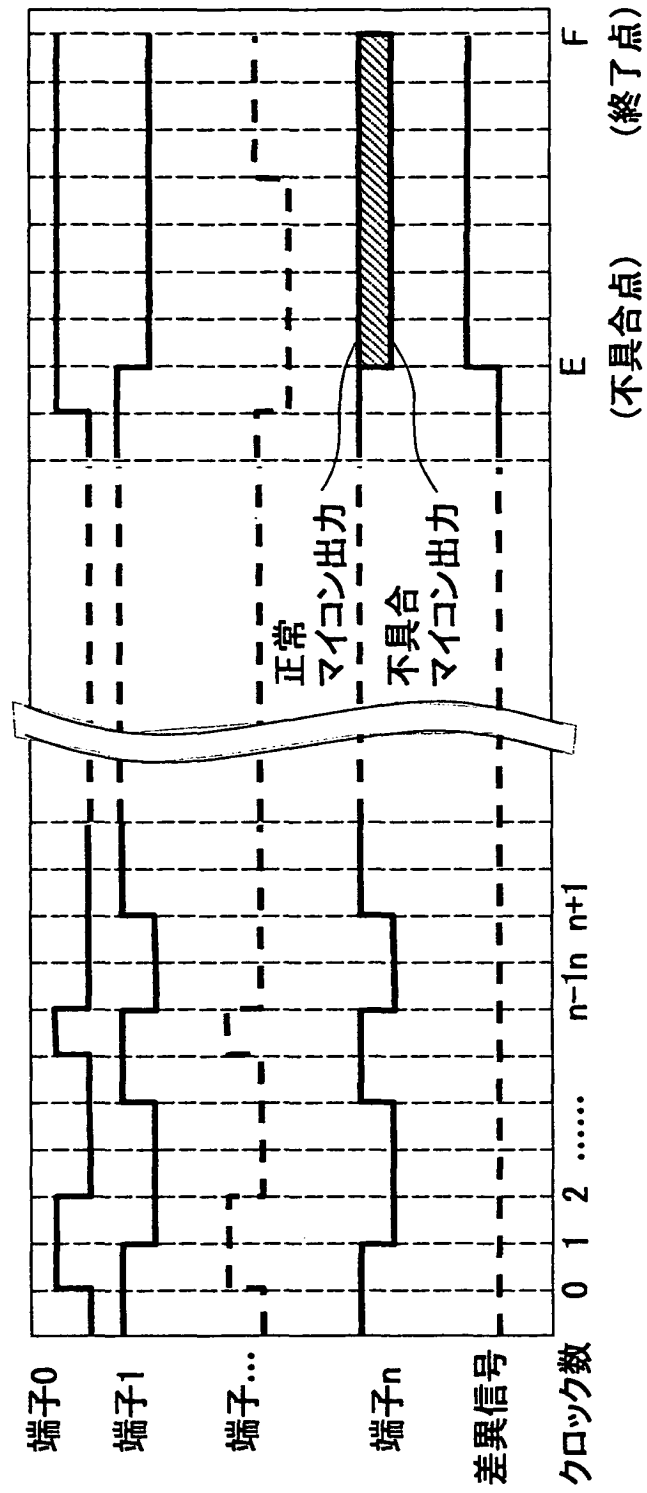




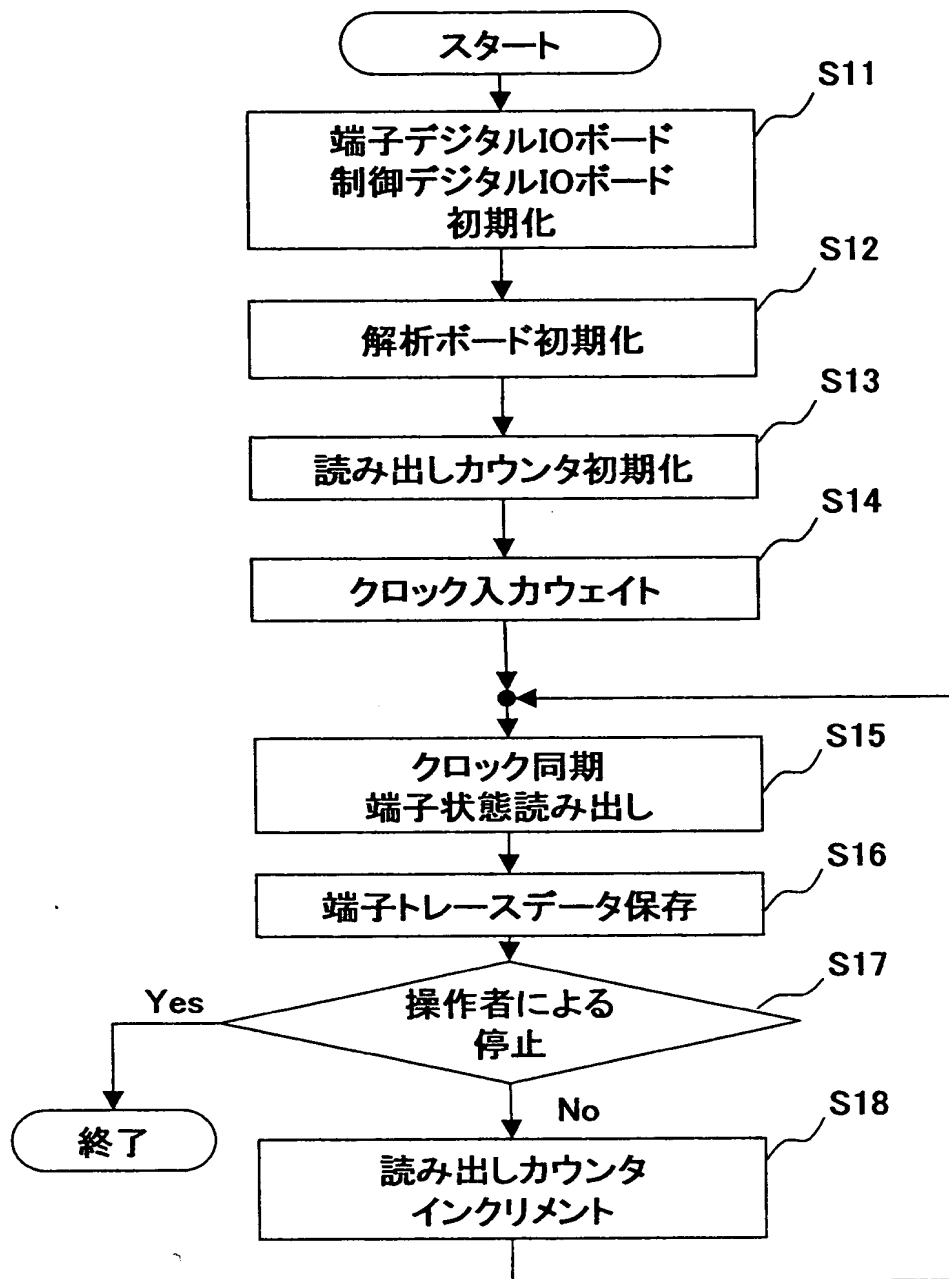
【図 8】



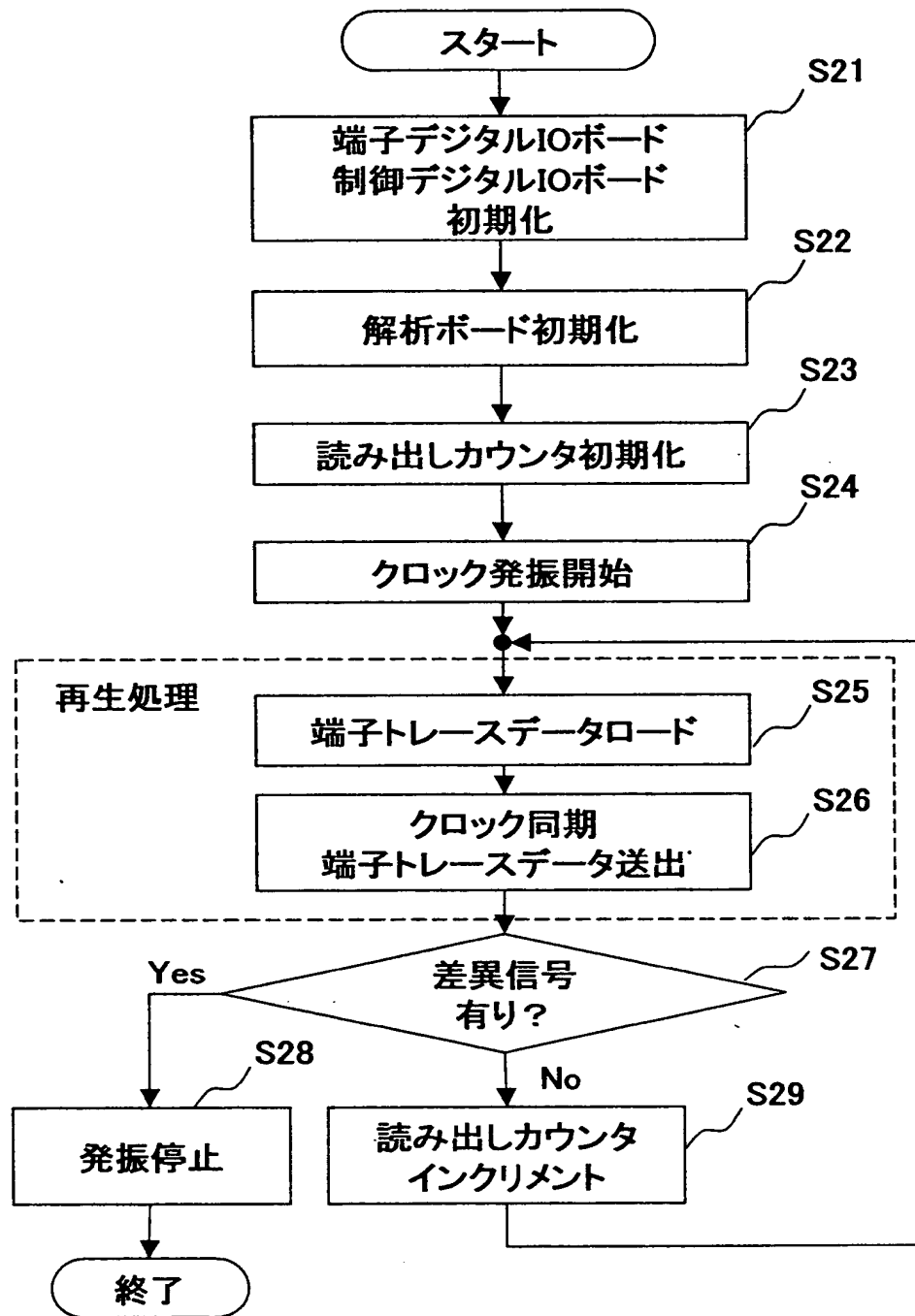
【図9】



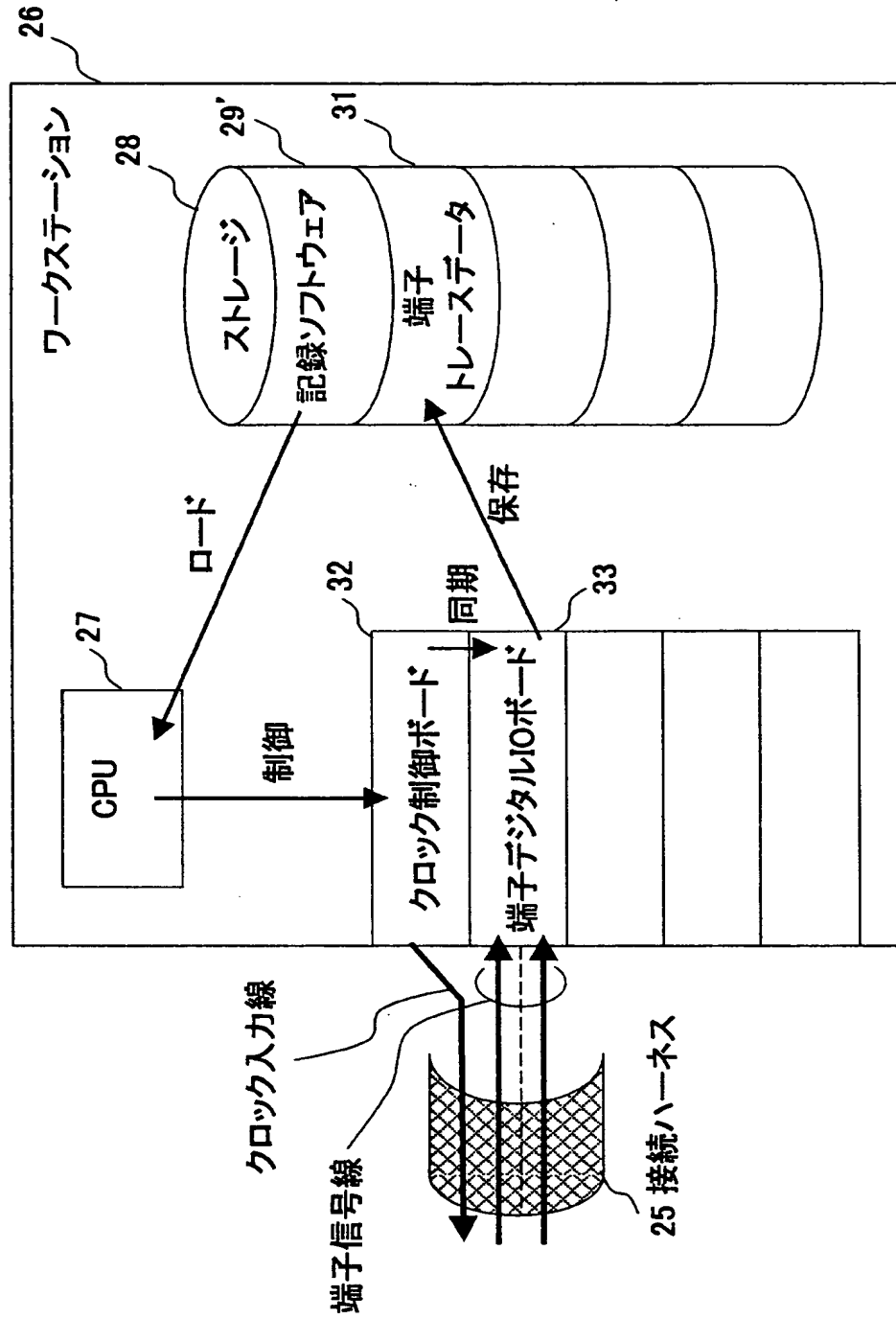
【図 1 0】



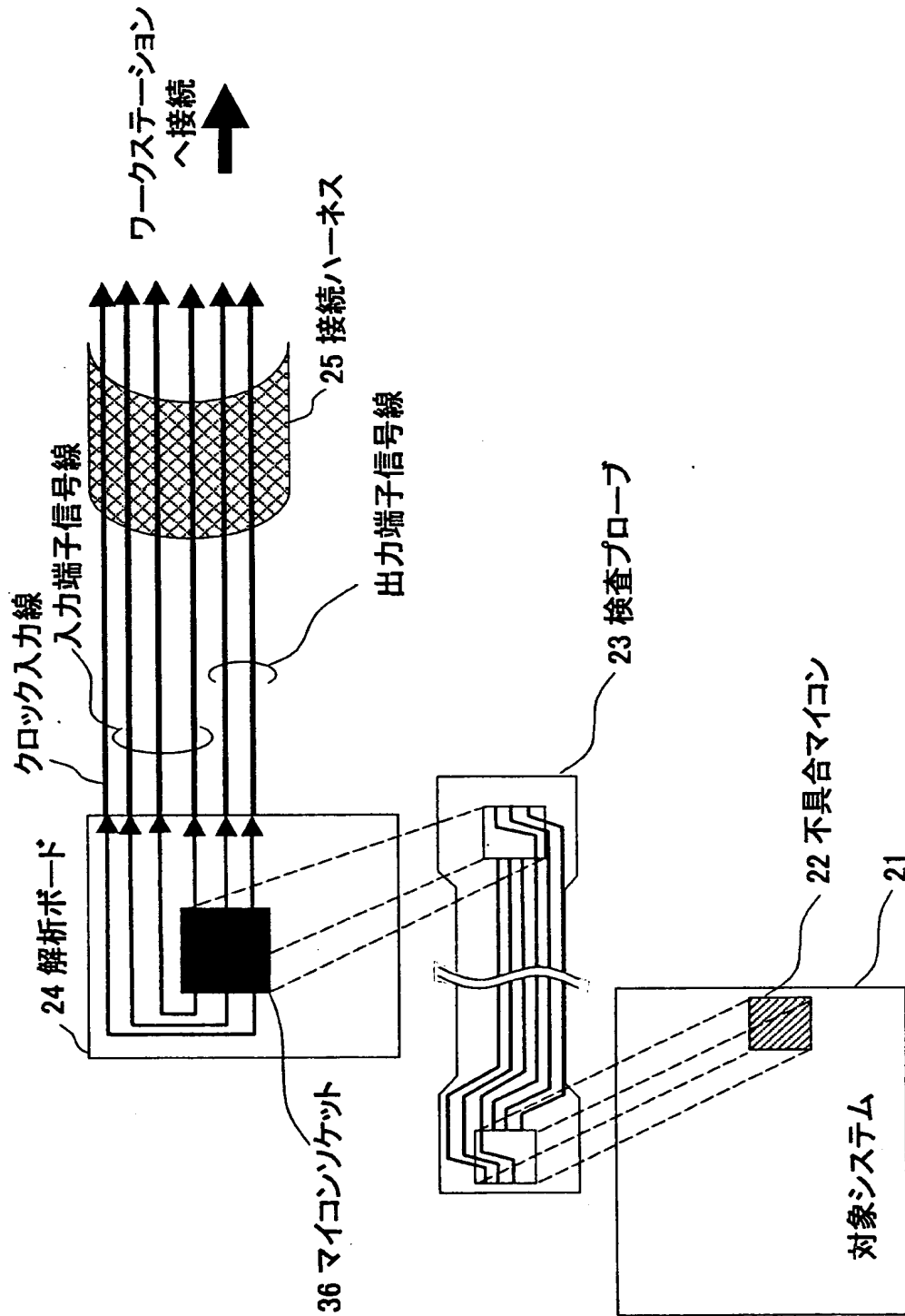
【図 11】



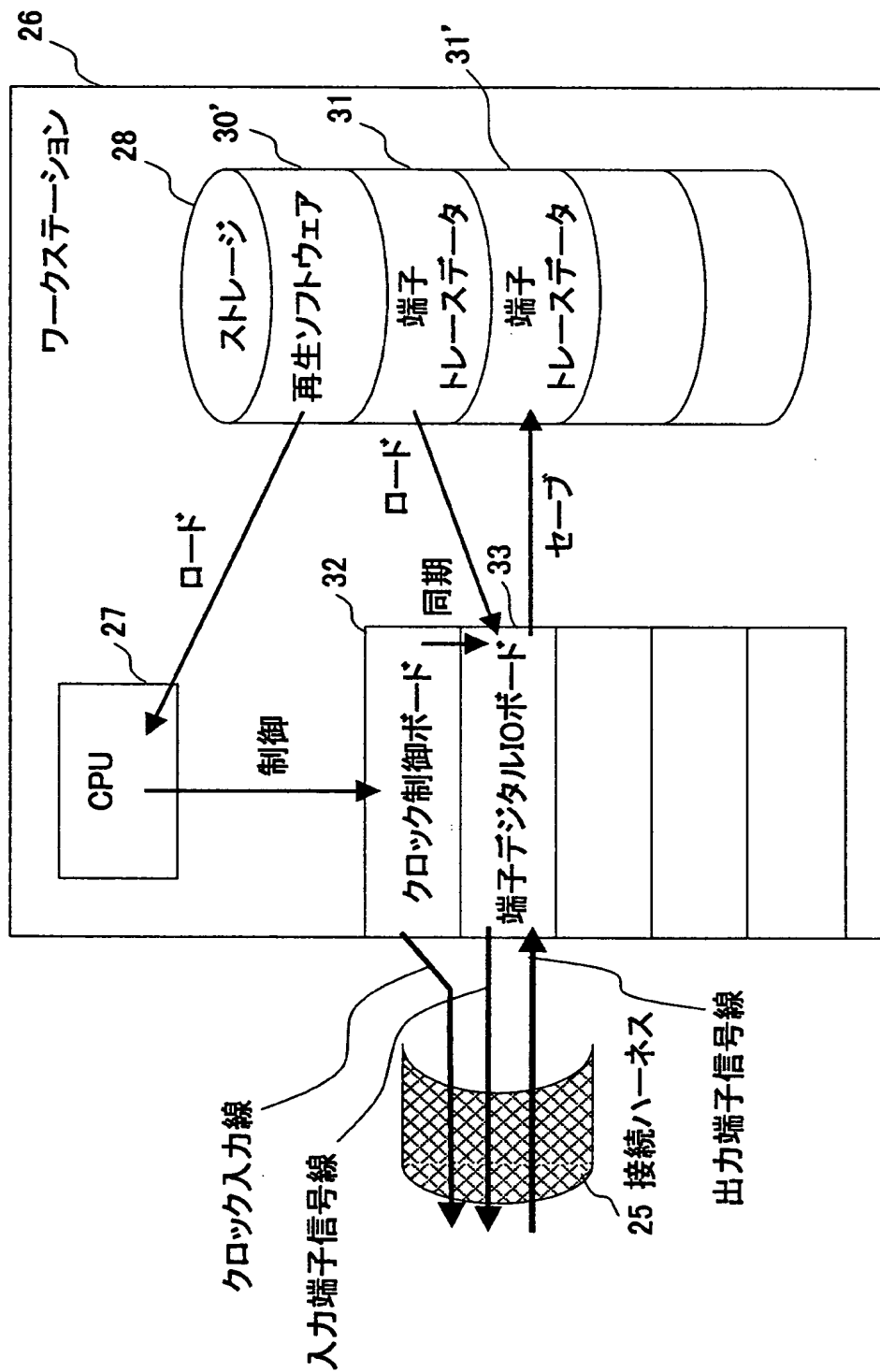
【図 12】



【図 13】

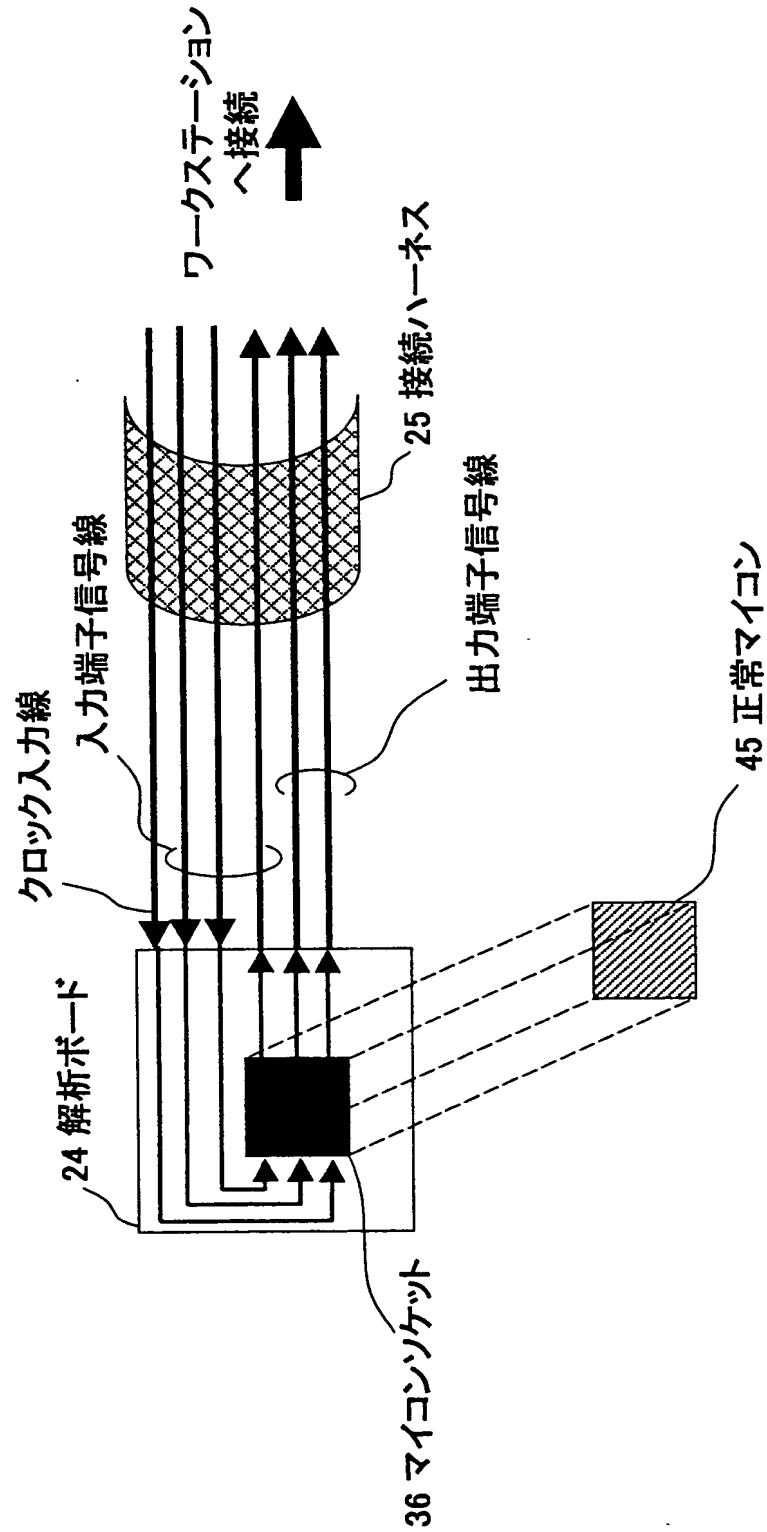


【図 14】

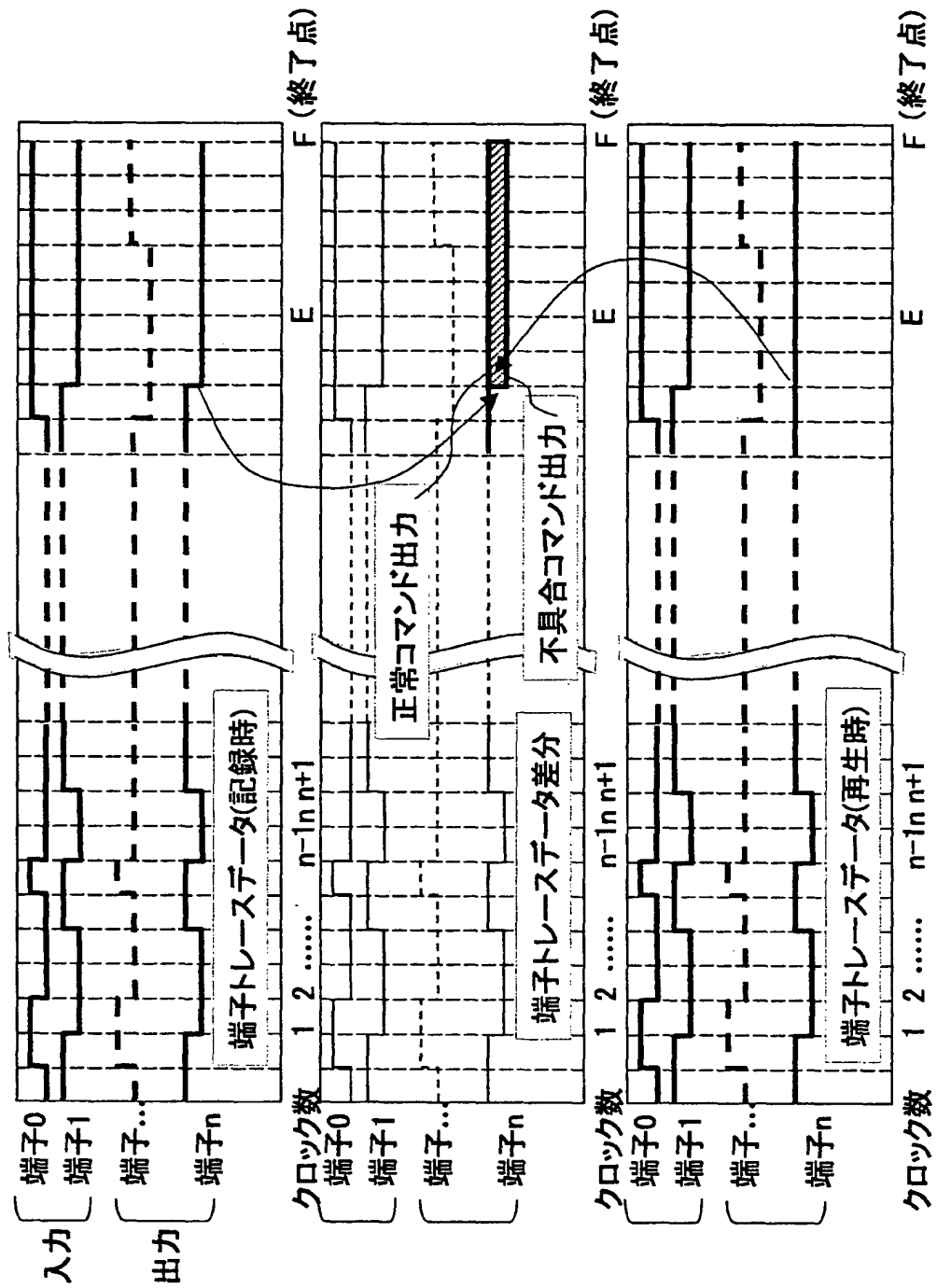




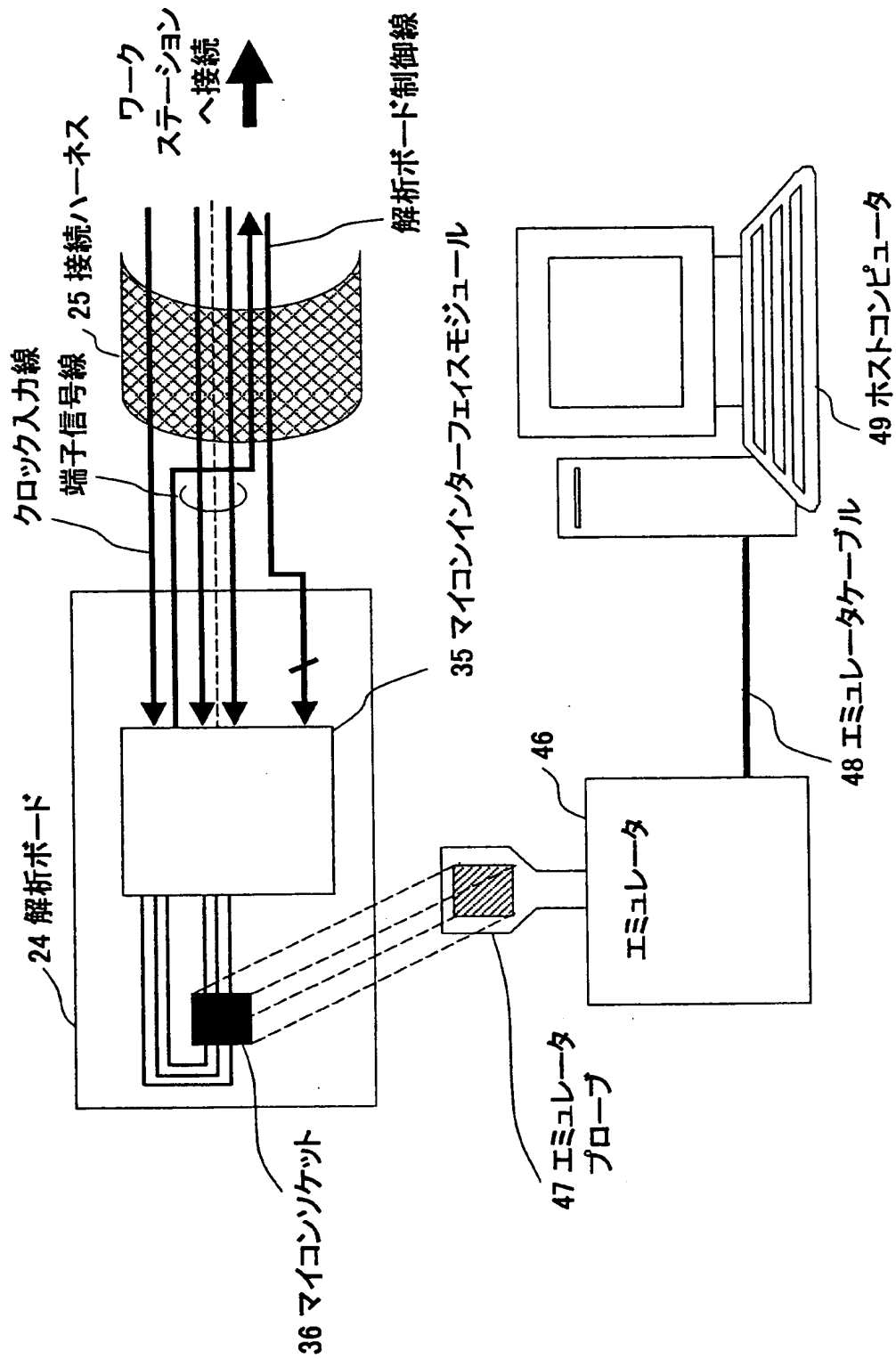
【図15】



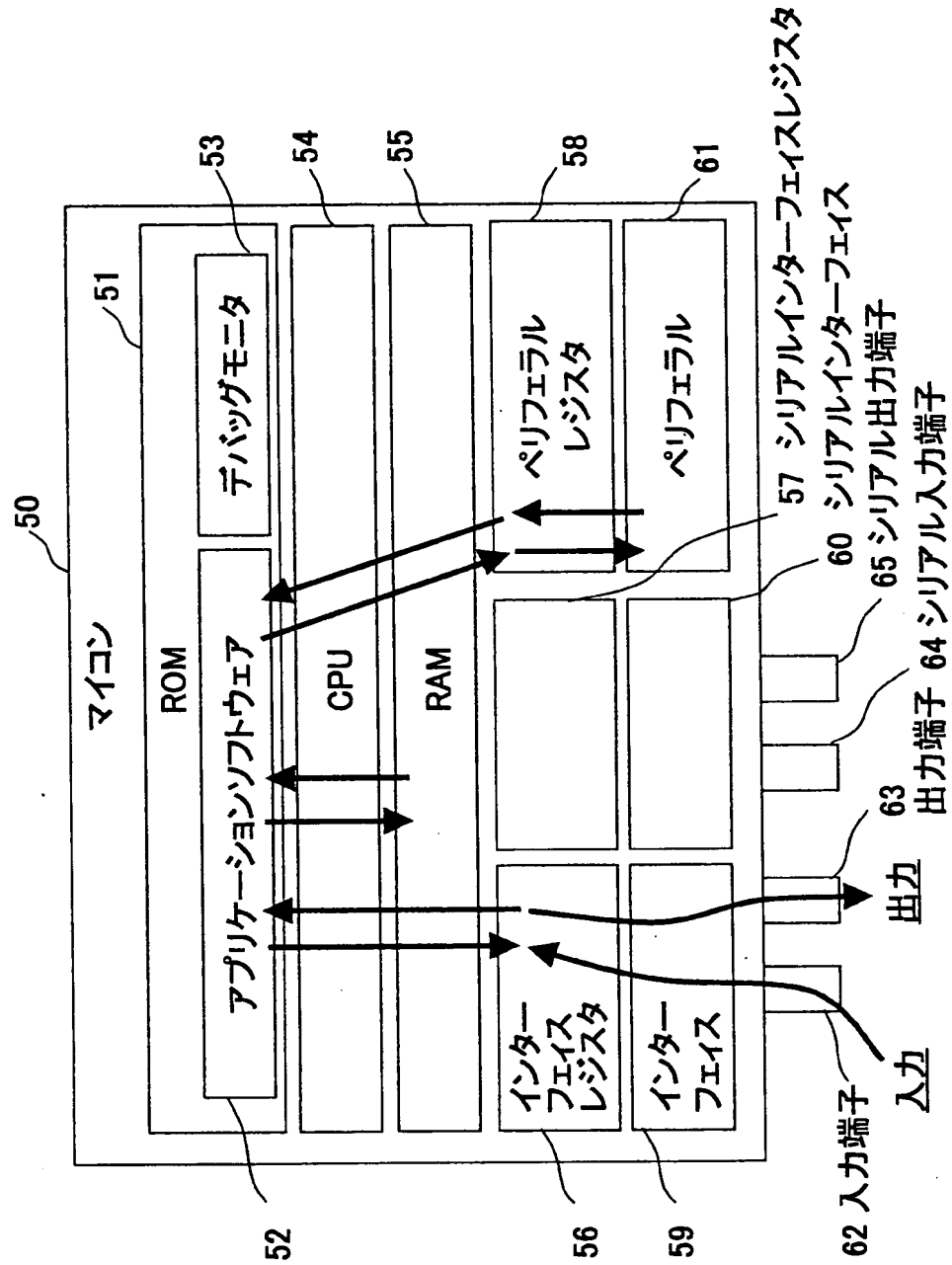
【図 16】



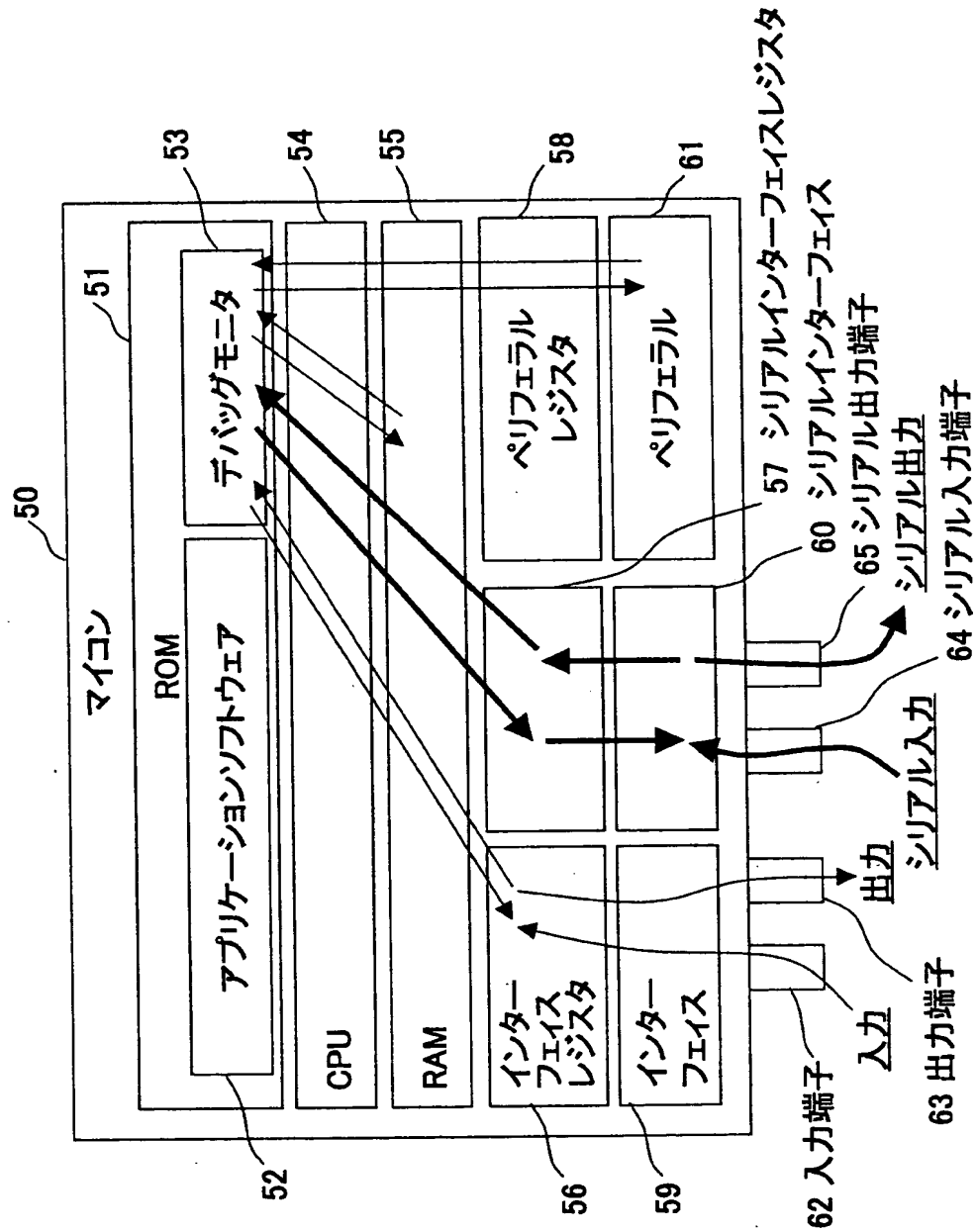
【図 17】



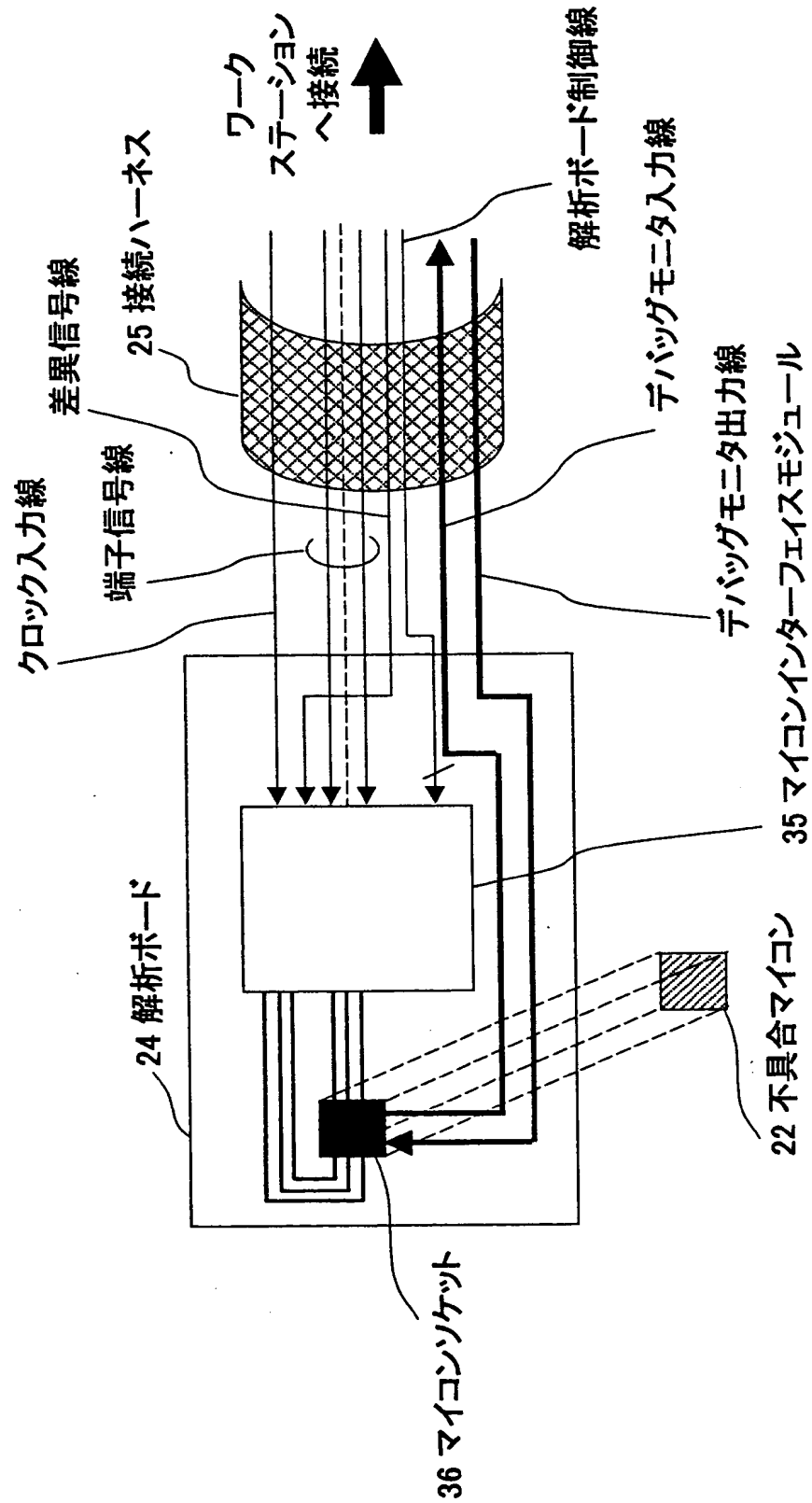
【図 18】



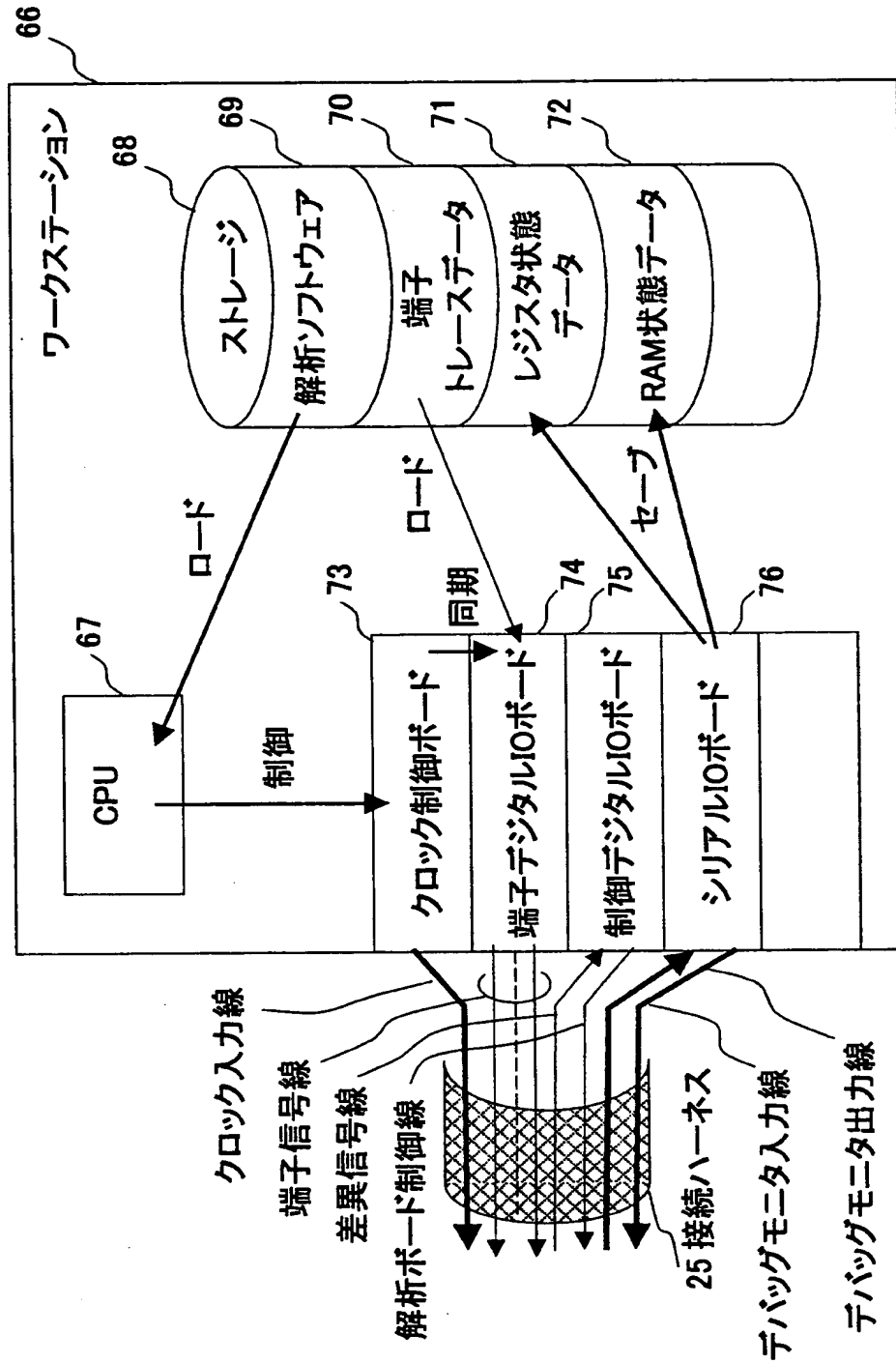
【図 19】



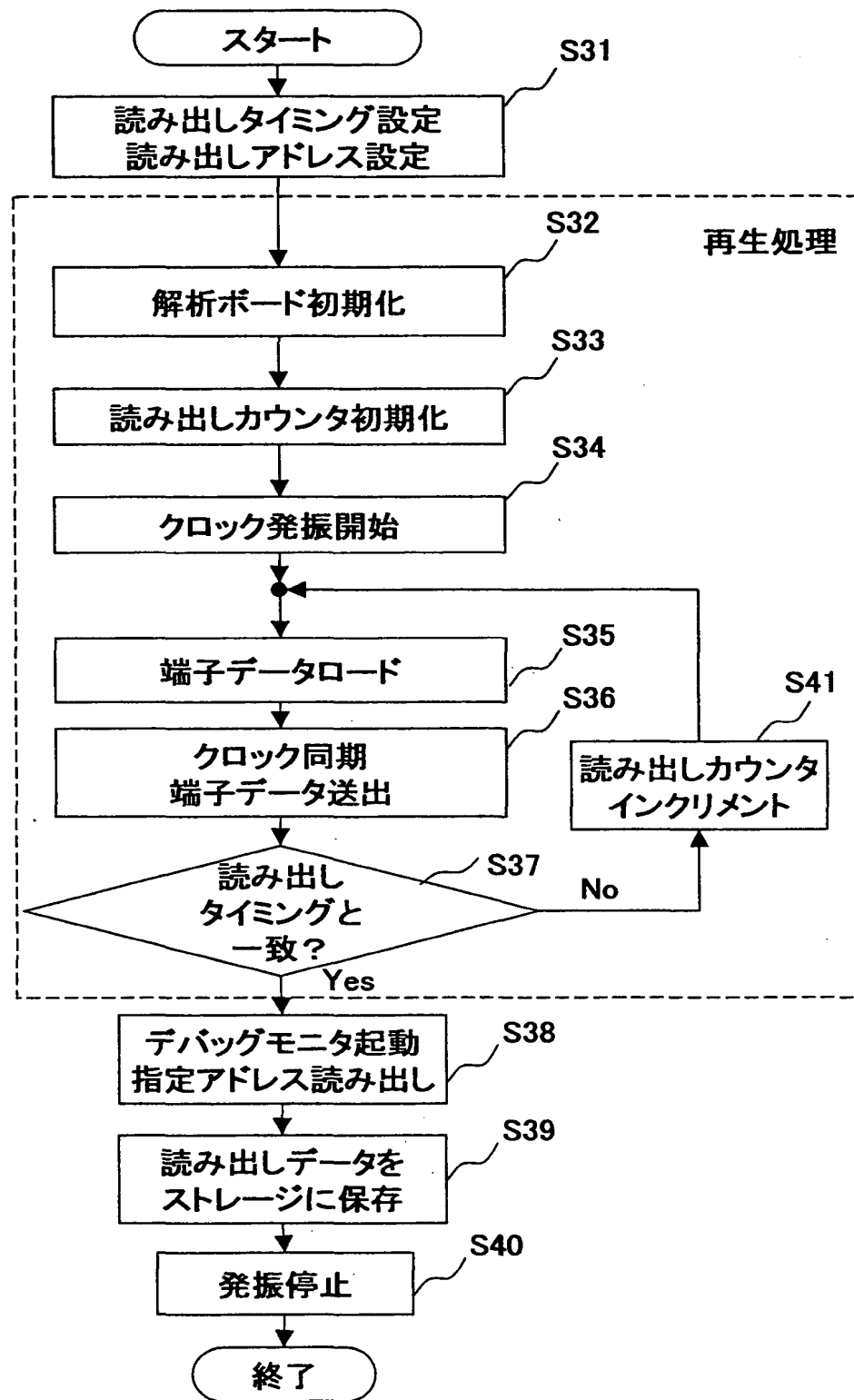
【図 20】



【図 21】

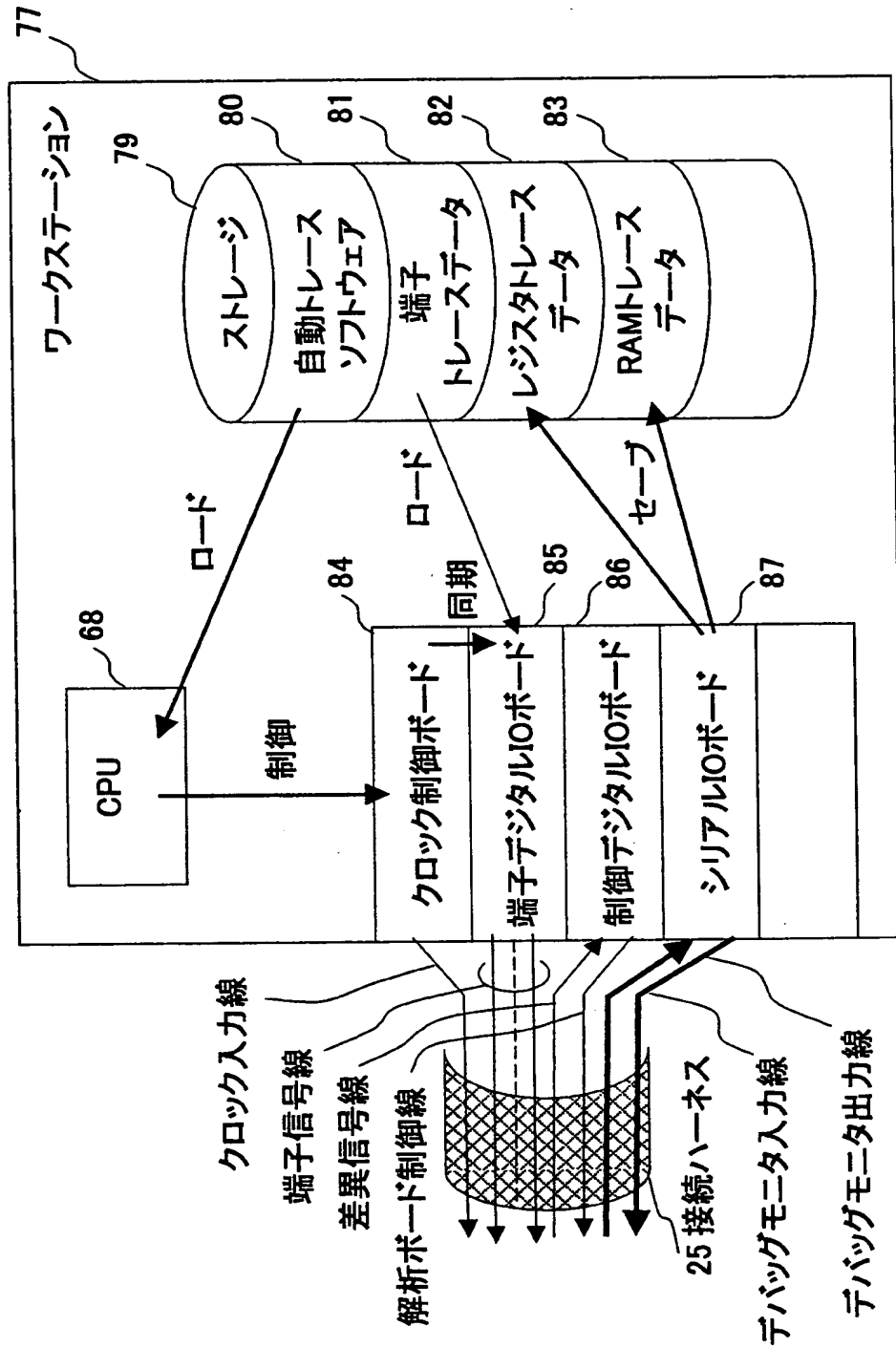


【図 22】

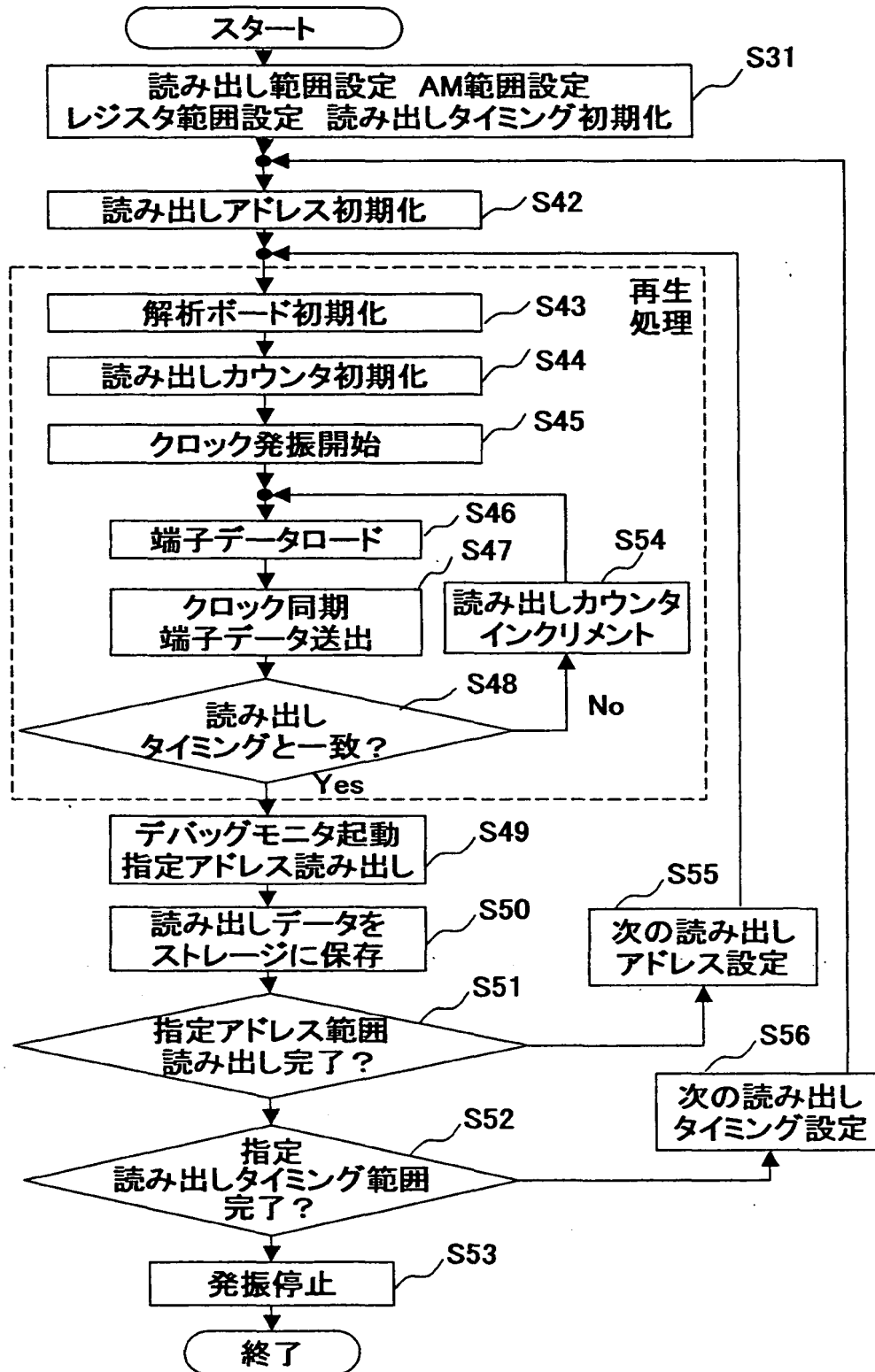




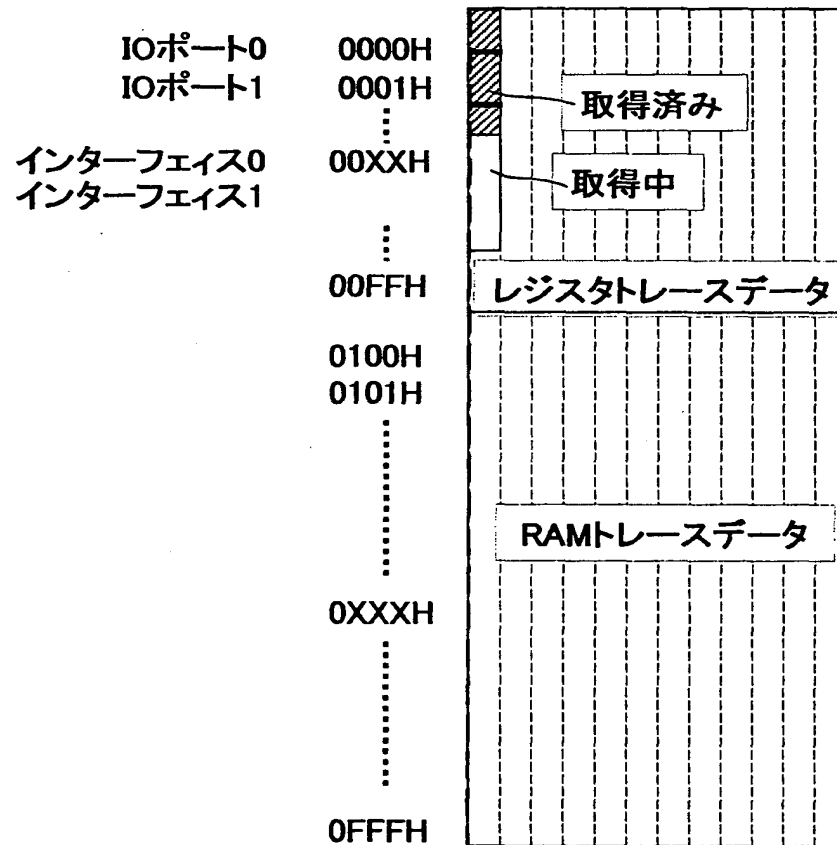
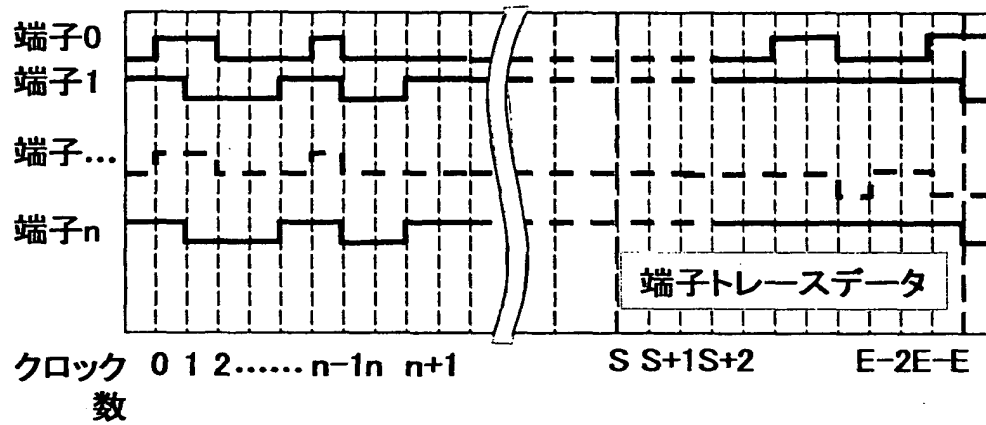
【図 23】



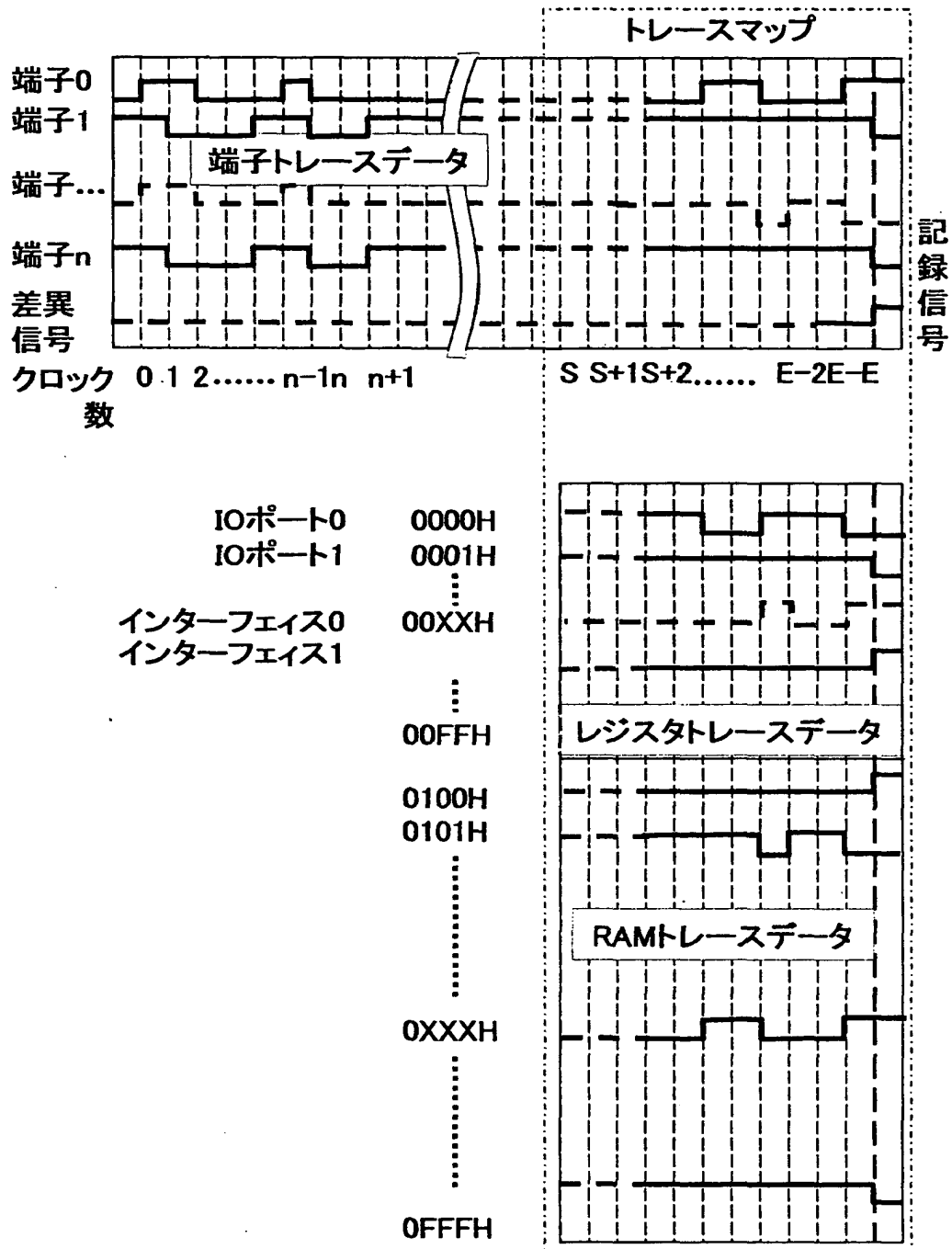
【図 24】



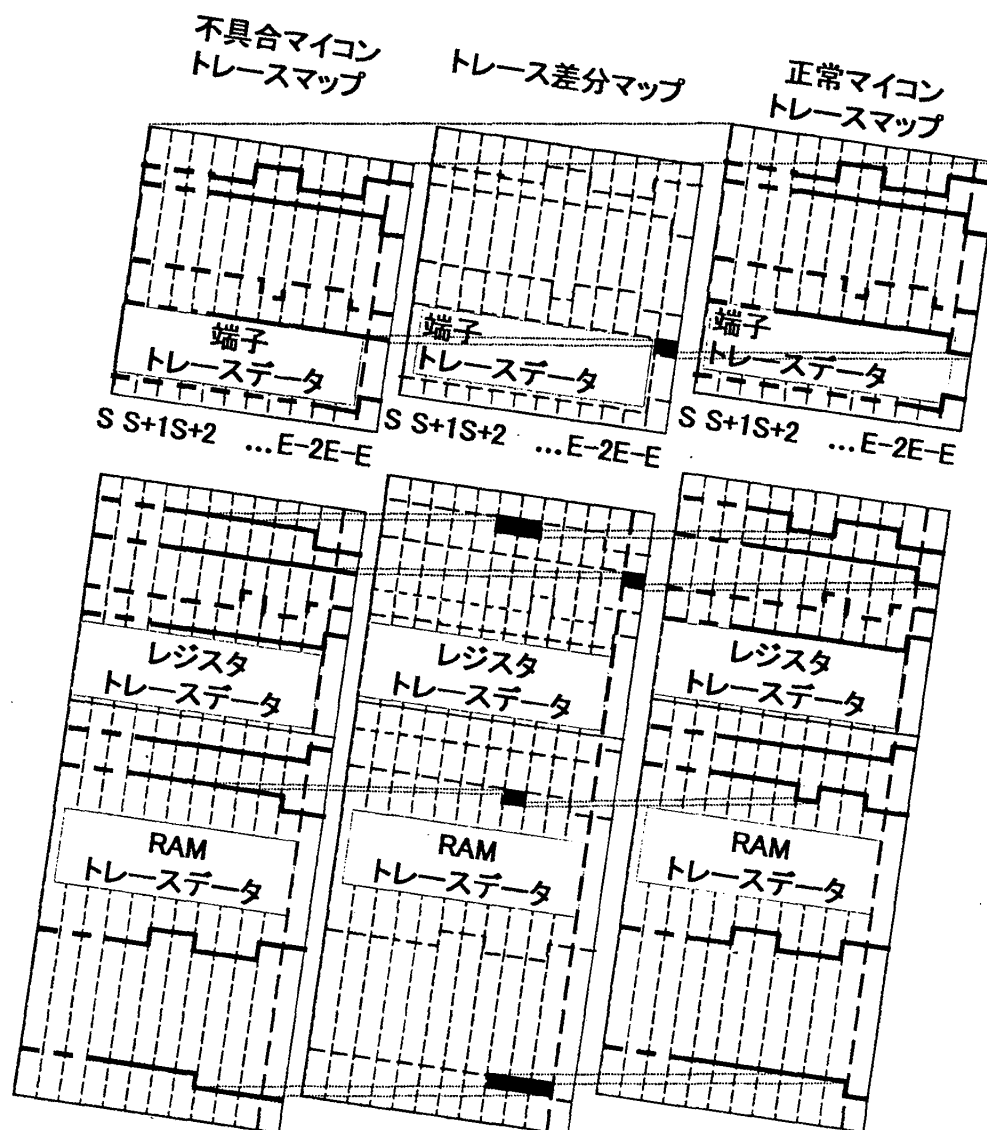
【図 2 5】



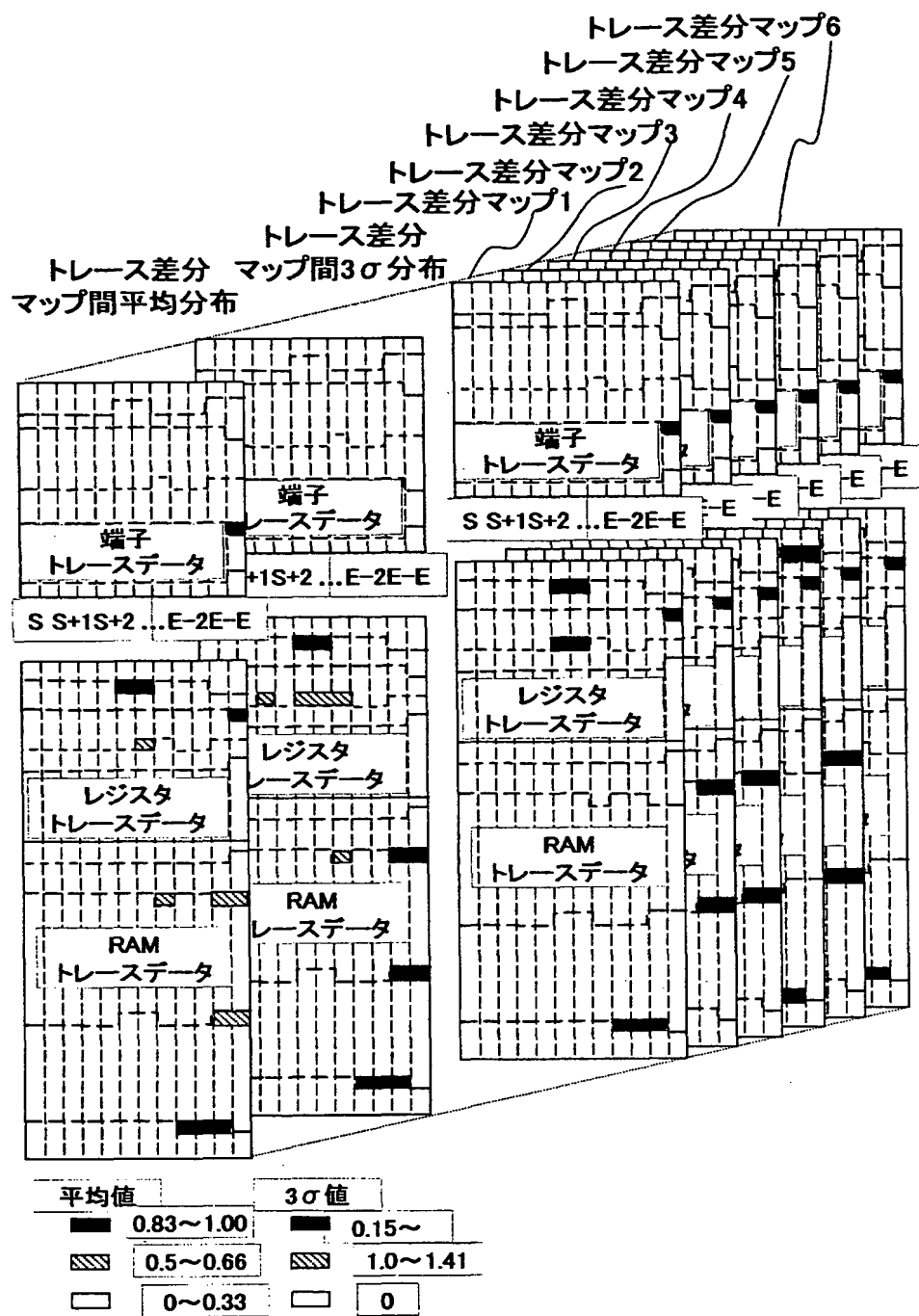
【図 26】



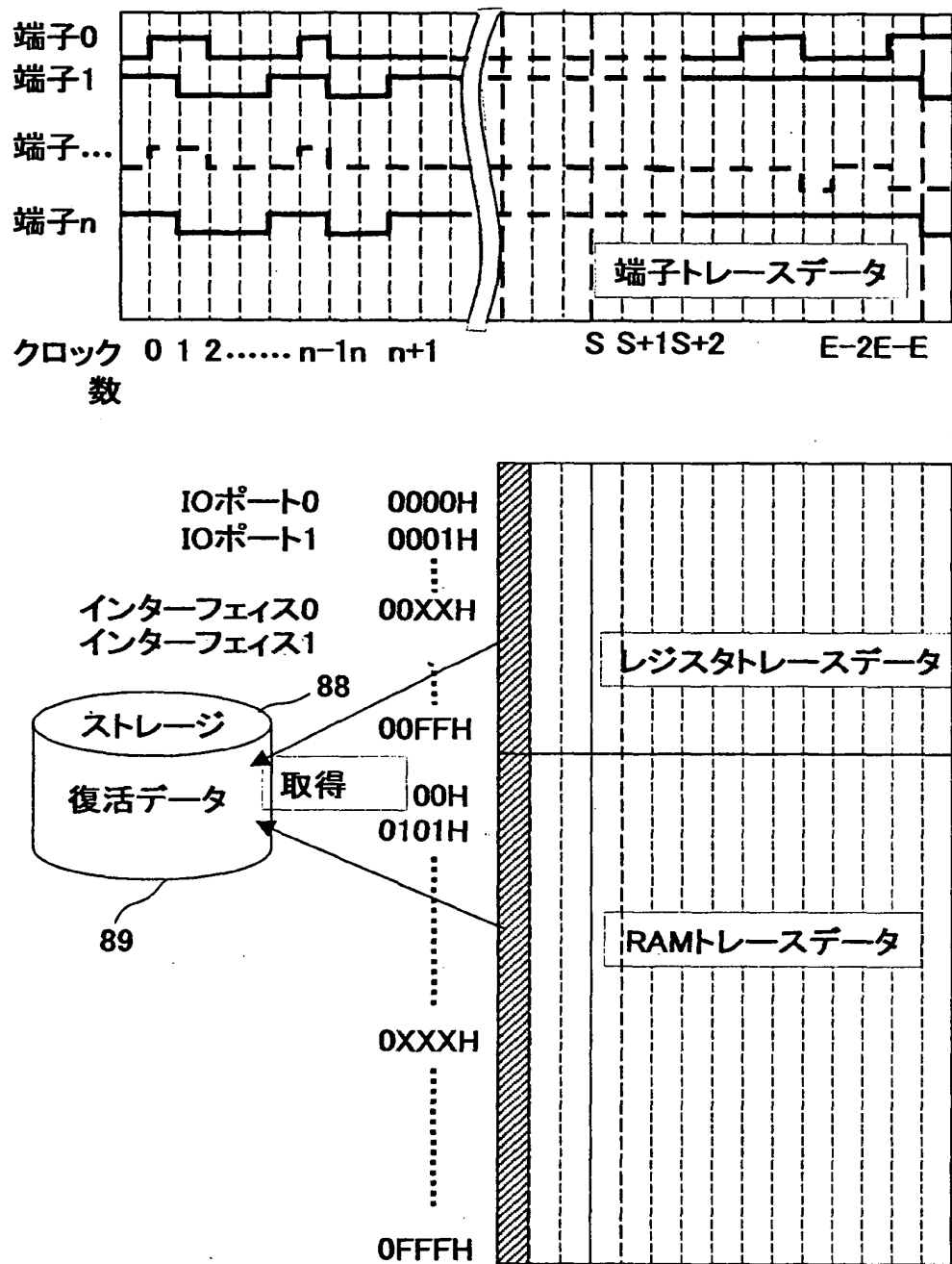
【図27】



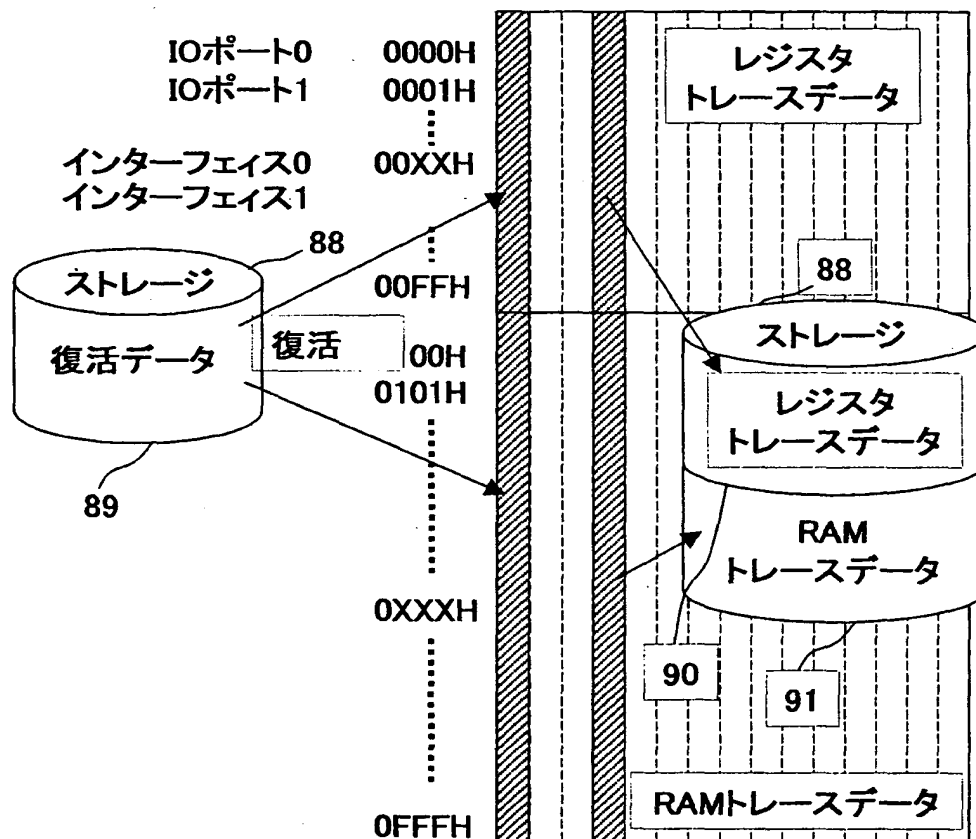
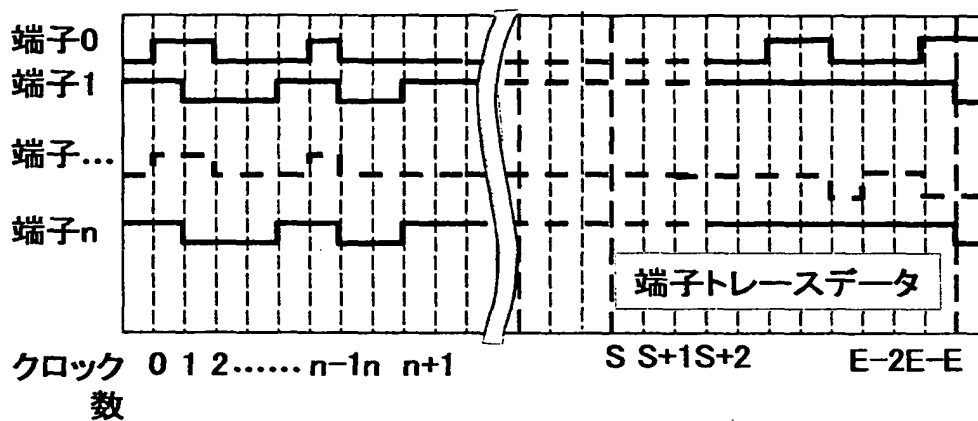
【图 28】



【図 2 9】

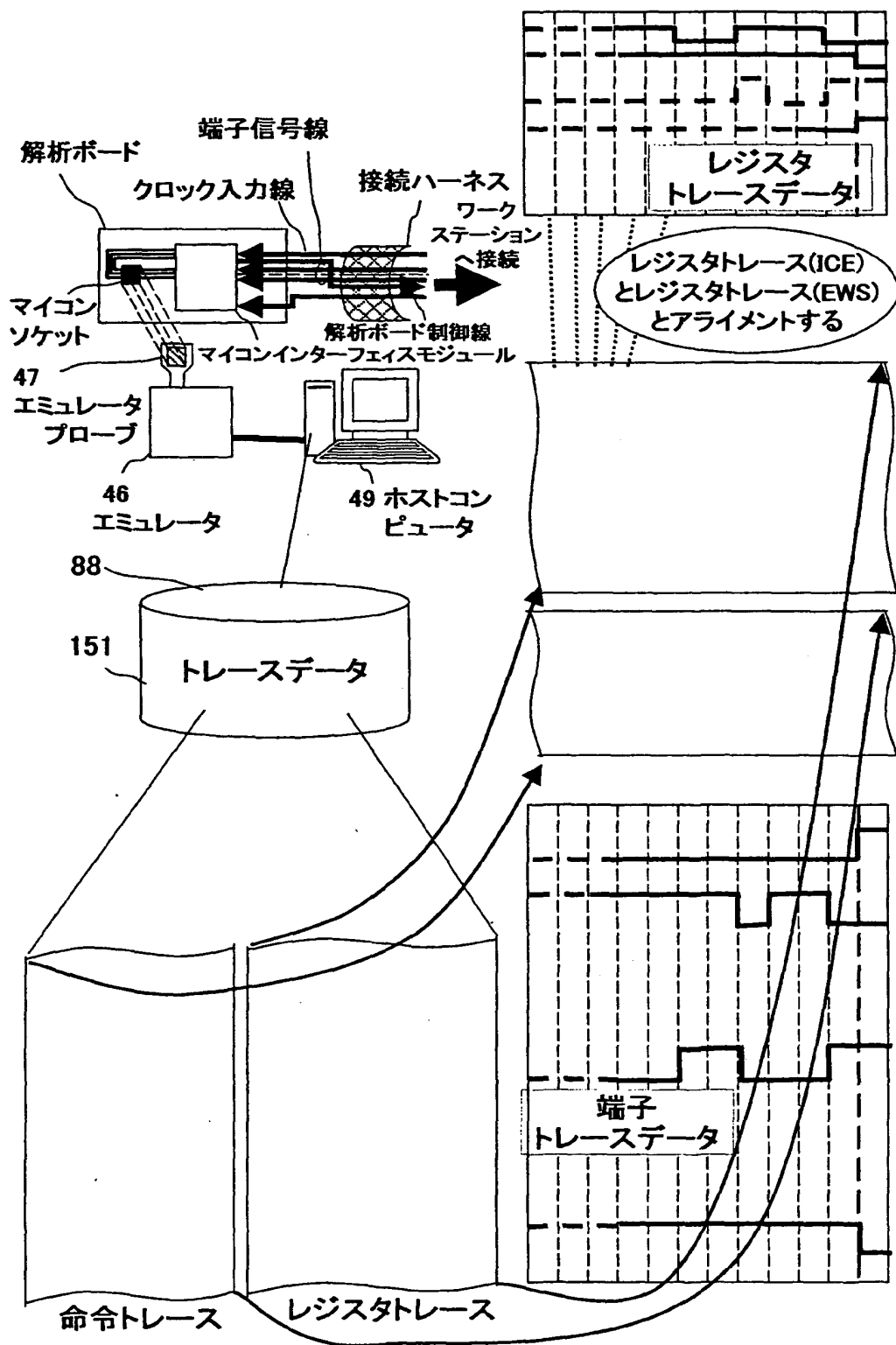


【図30】

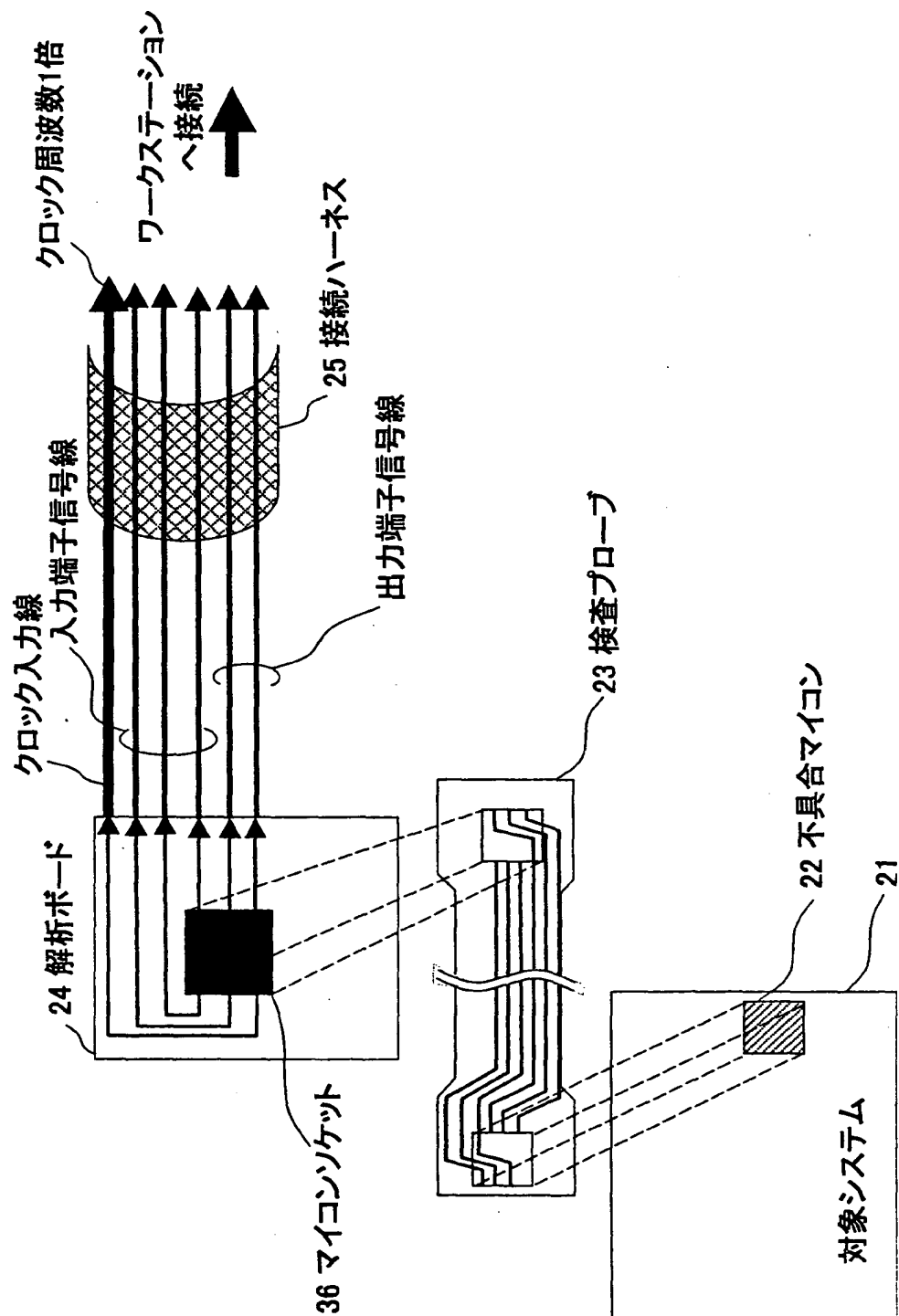




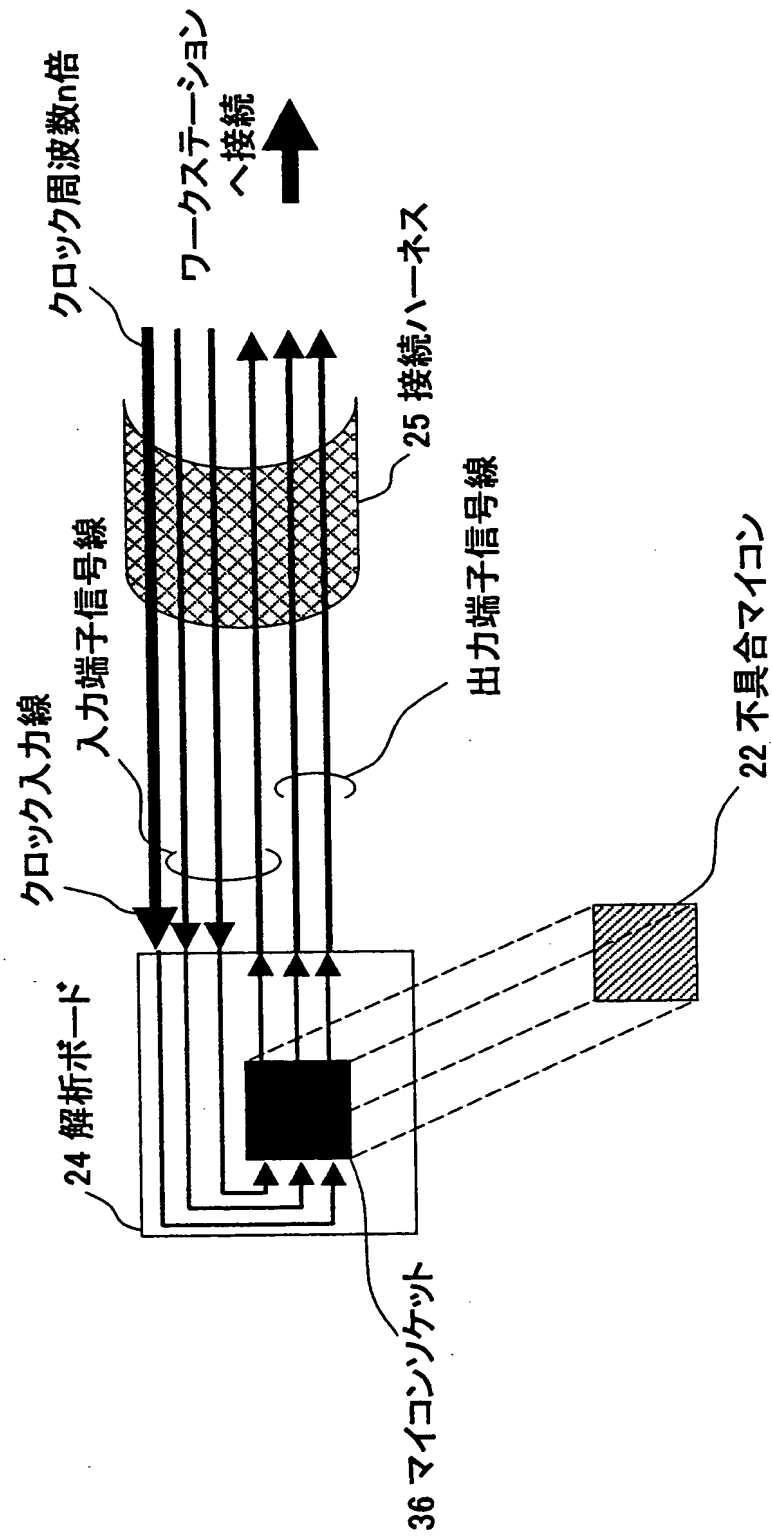
【図 31】



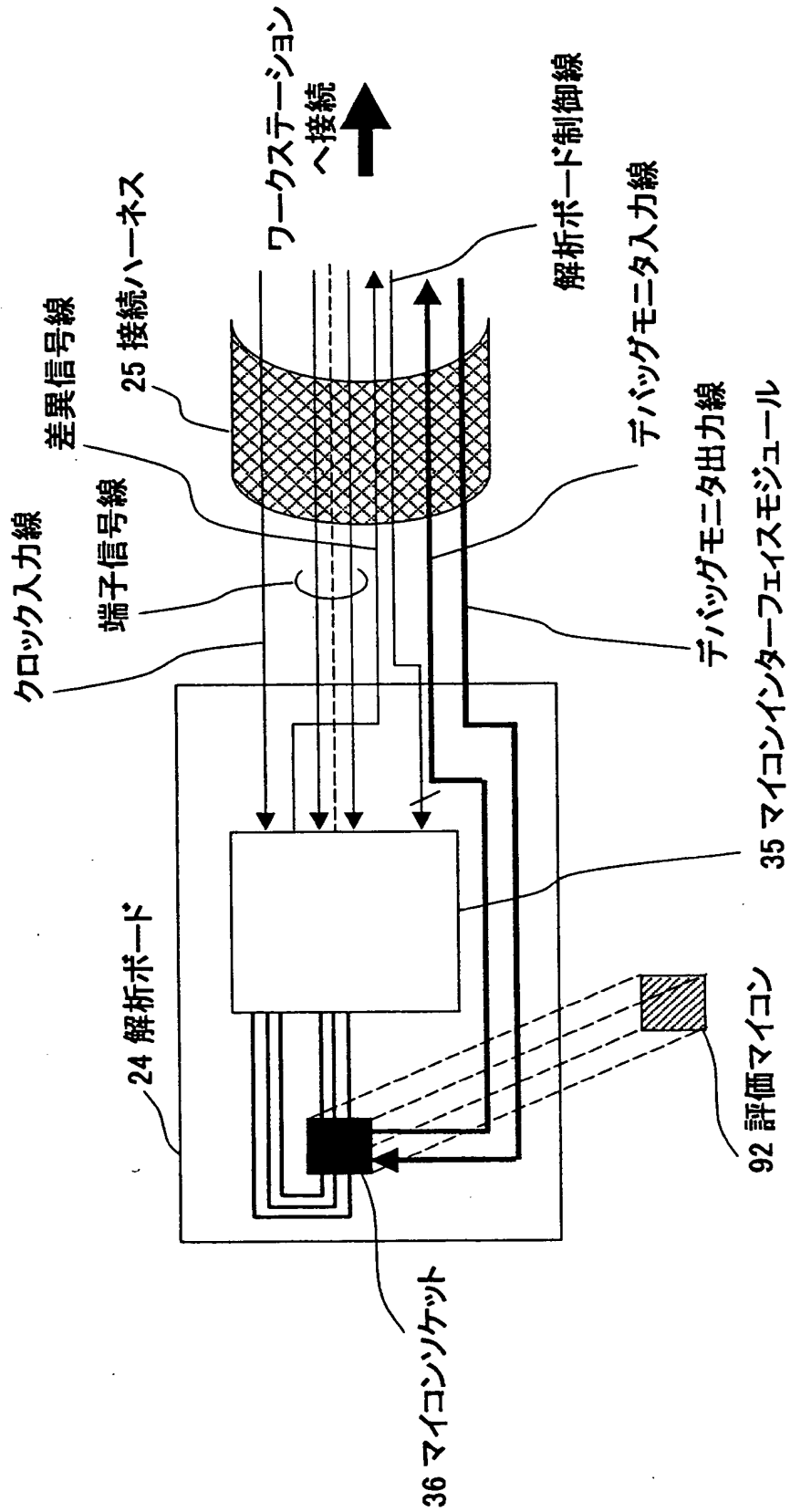
【図 32】



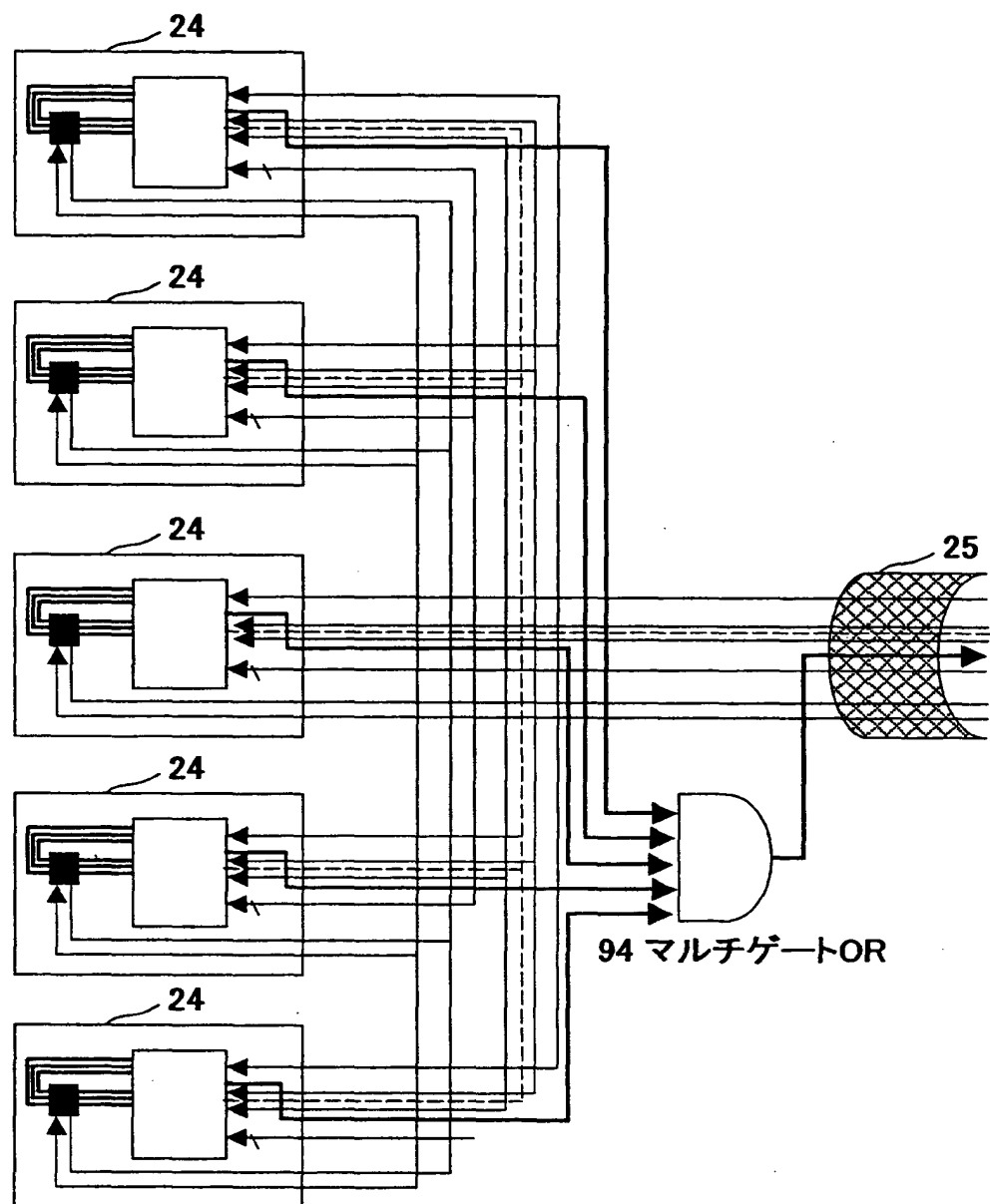
【図 33】



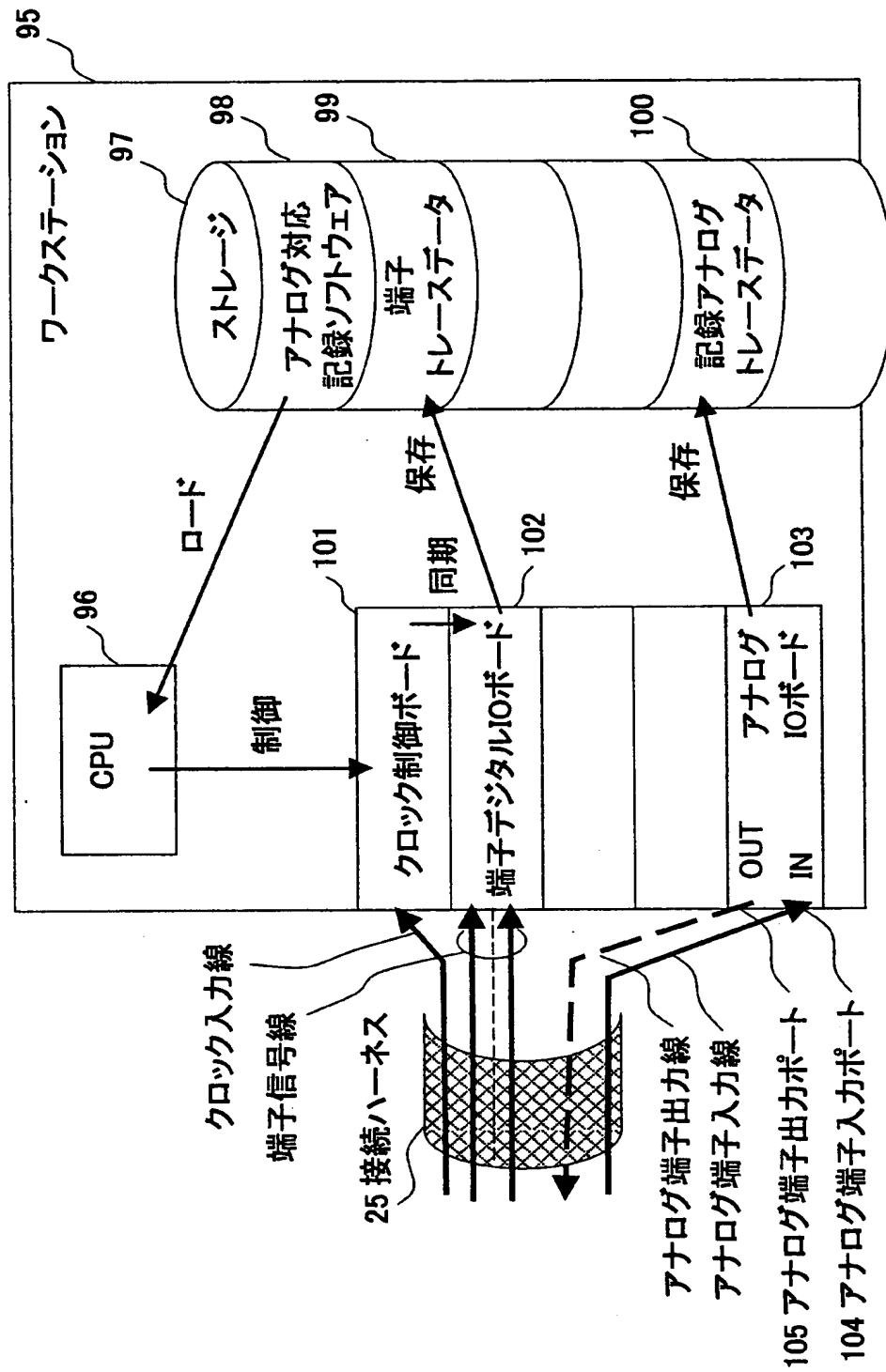
【図 3 4】



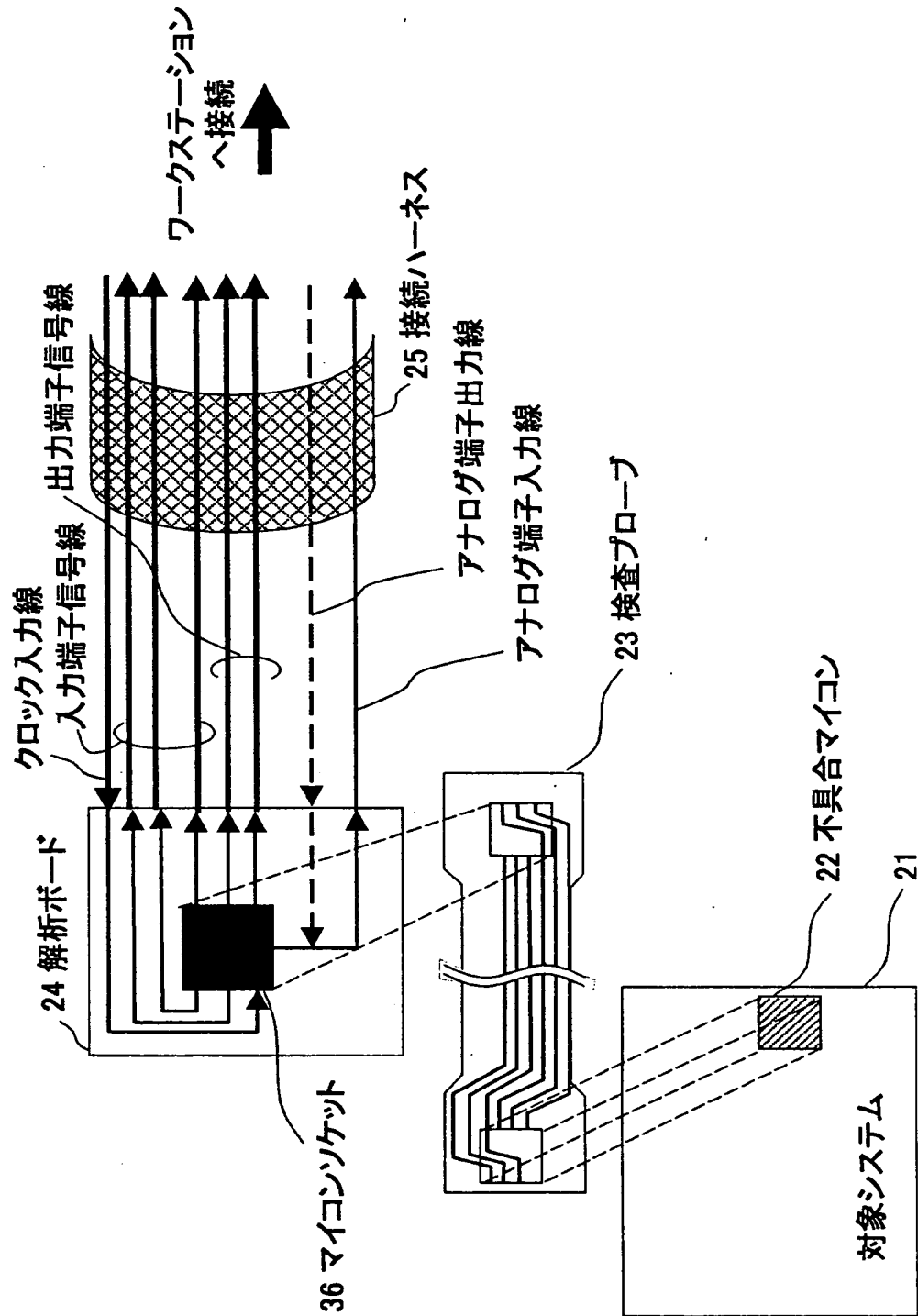
【図 3 5】



【図36】

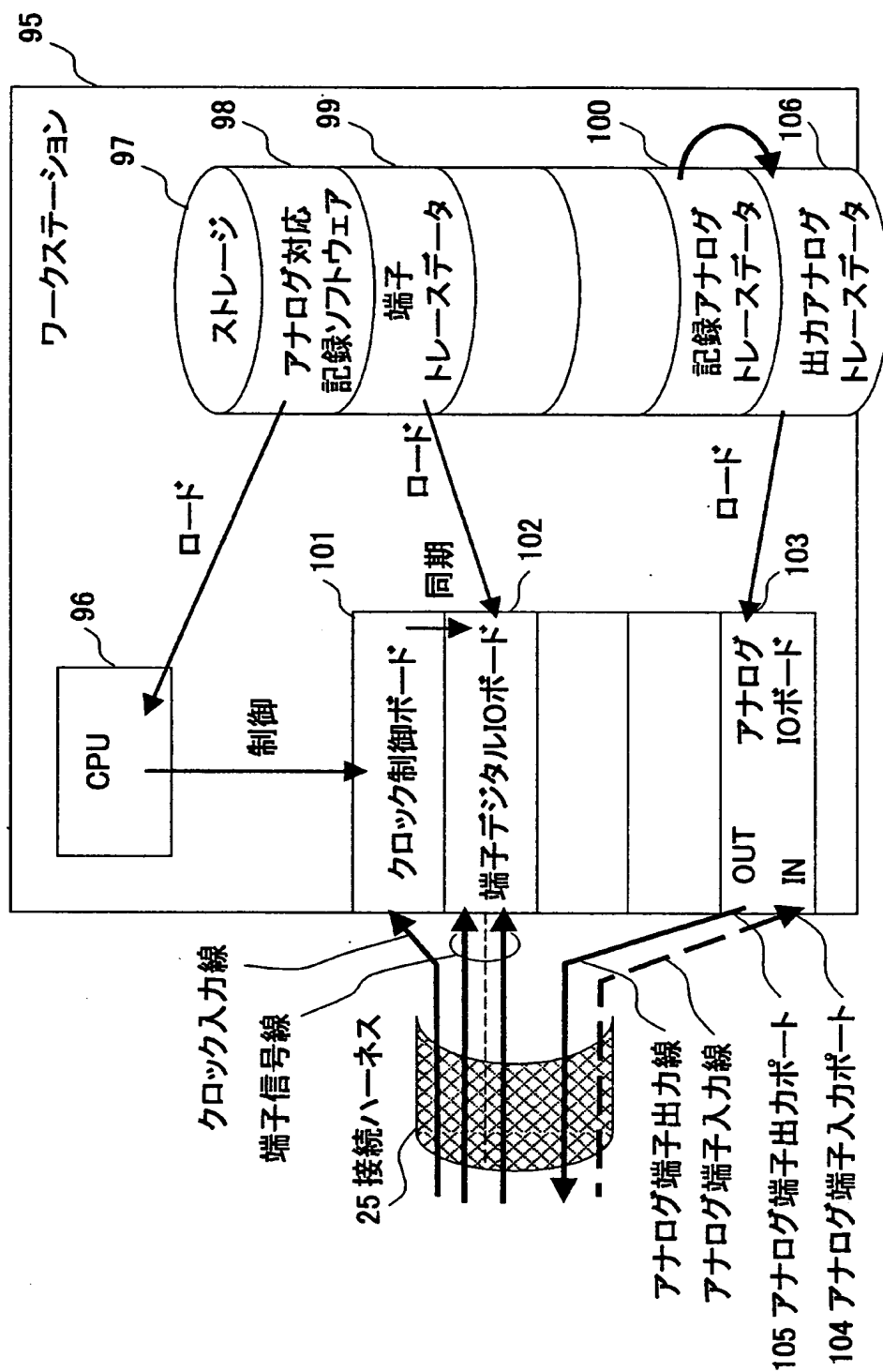


【図 37】

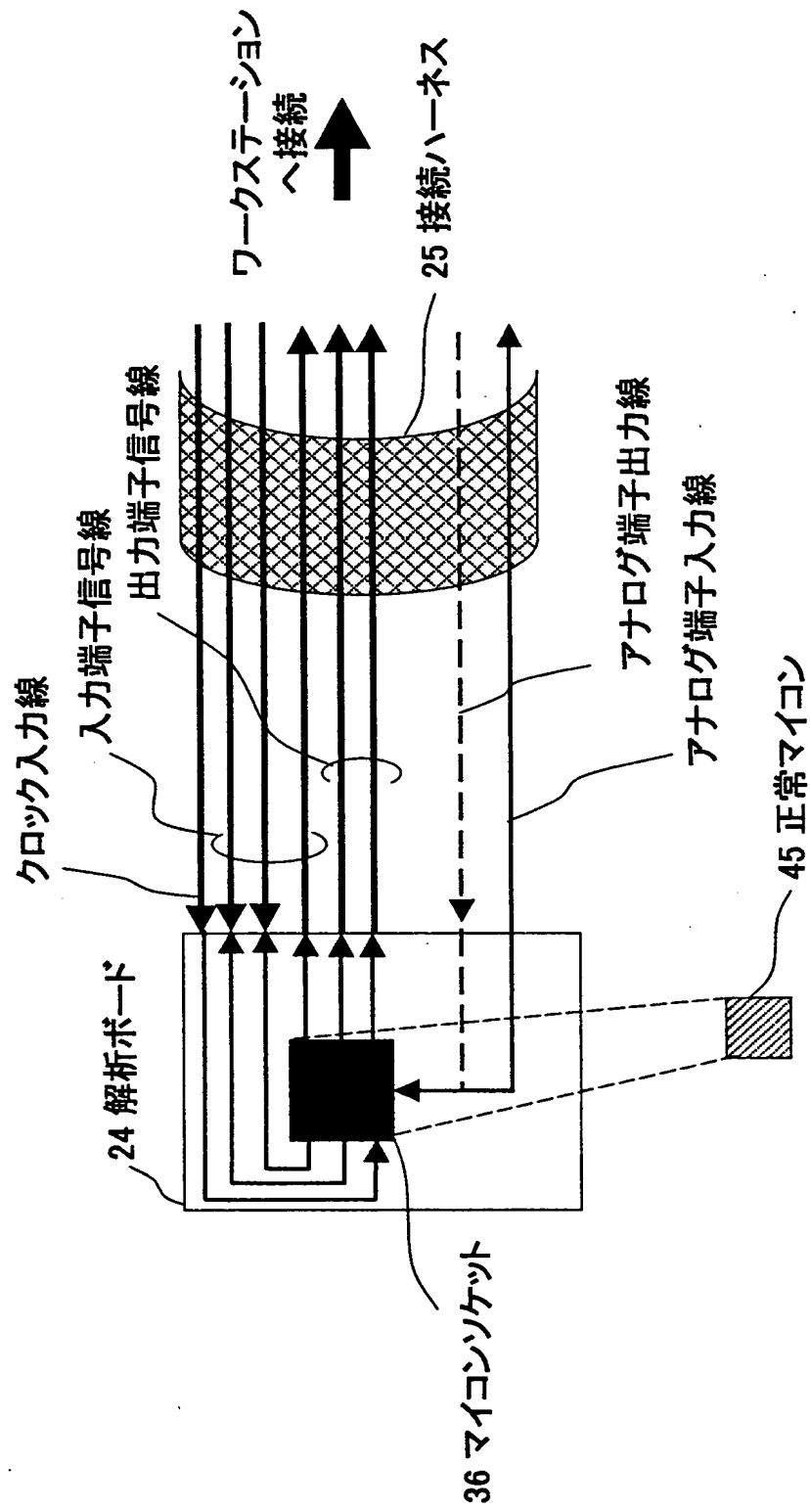




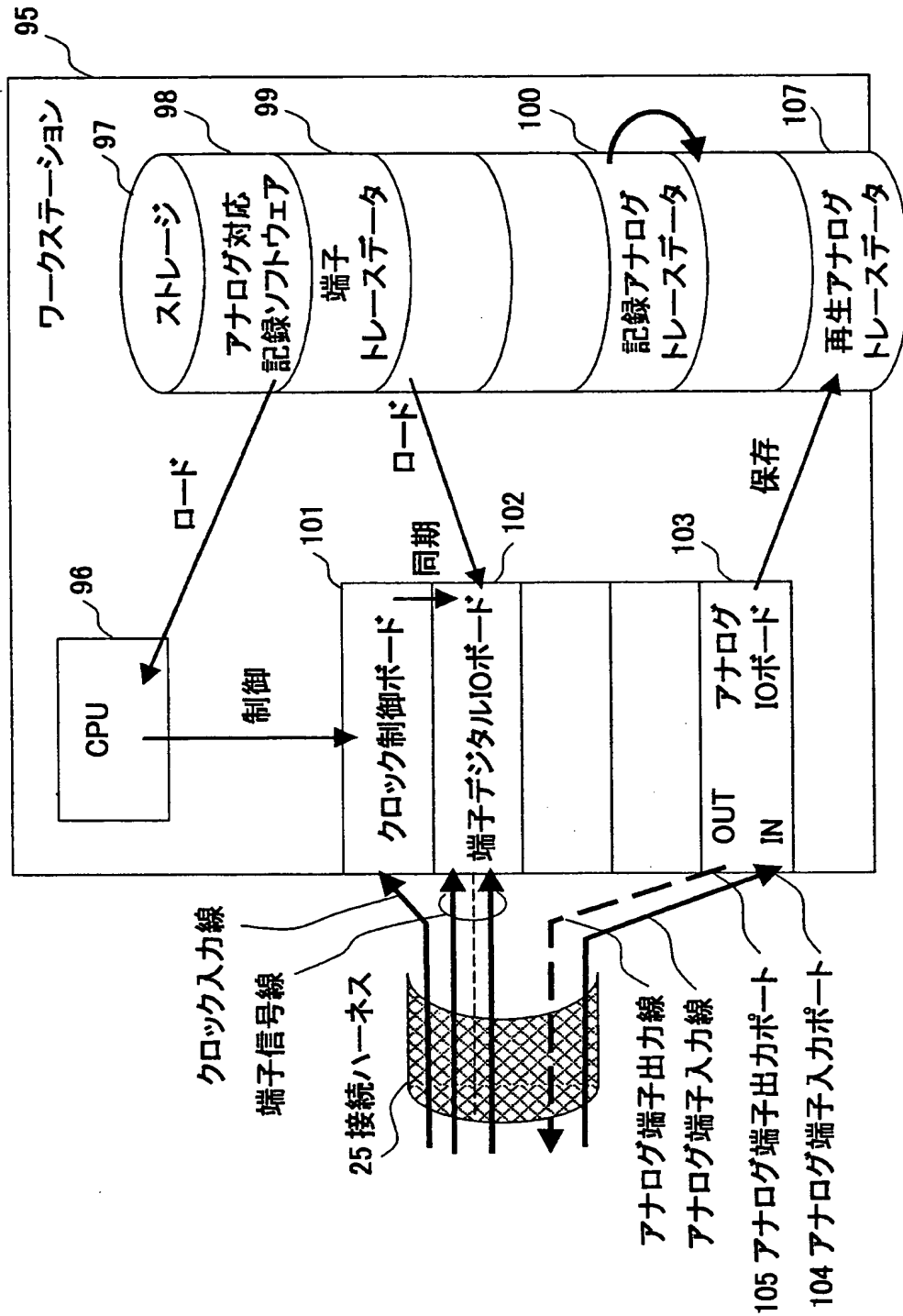
【図 38】



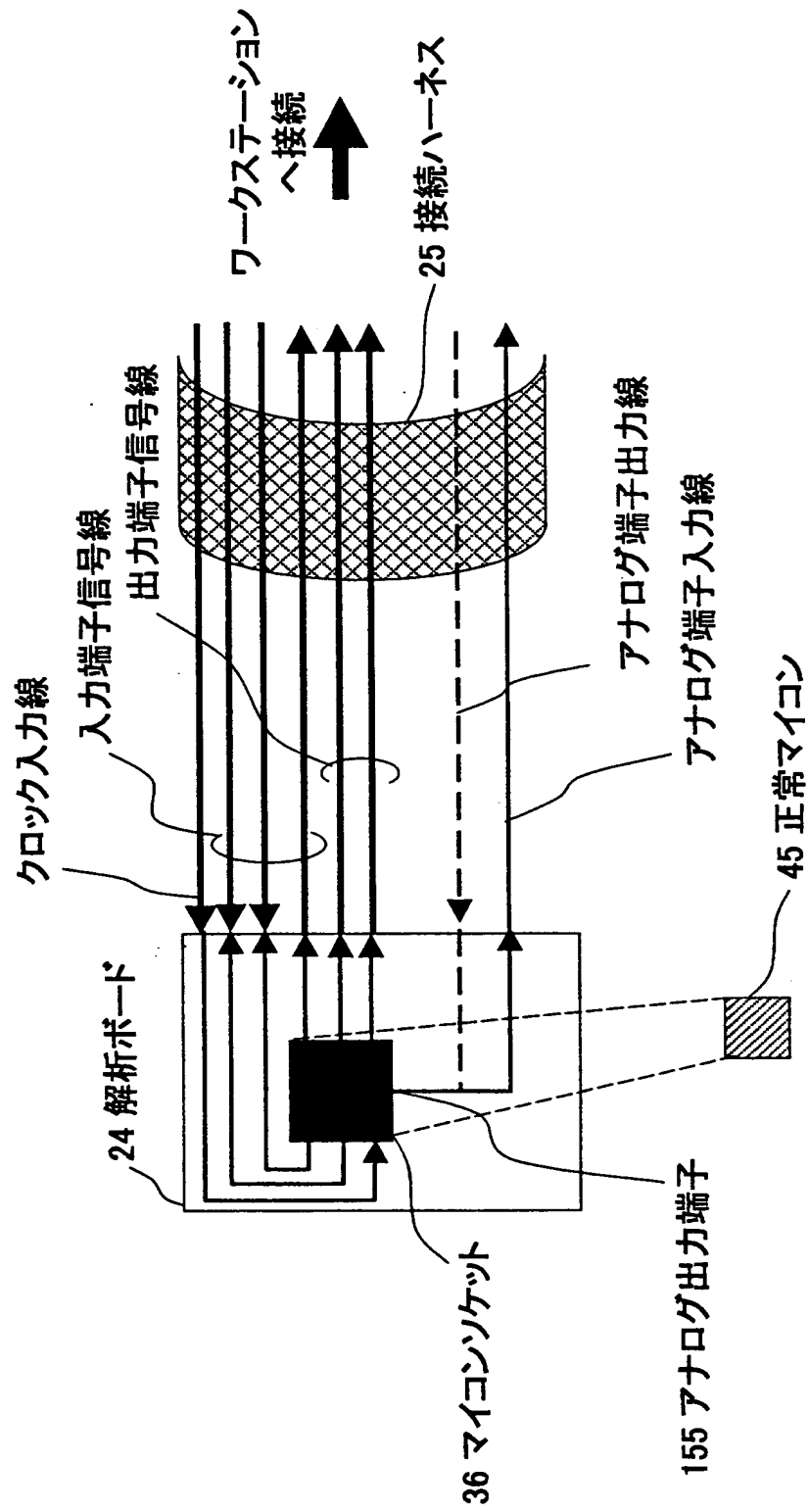
【図 39】



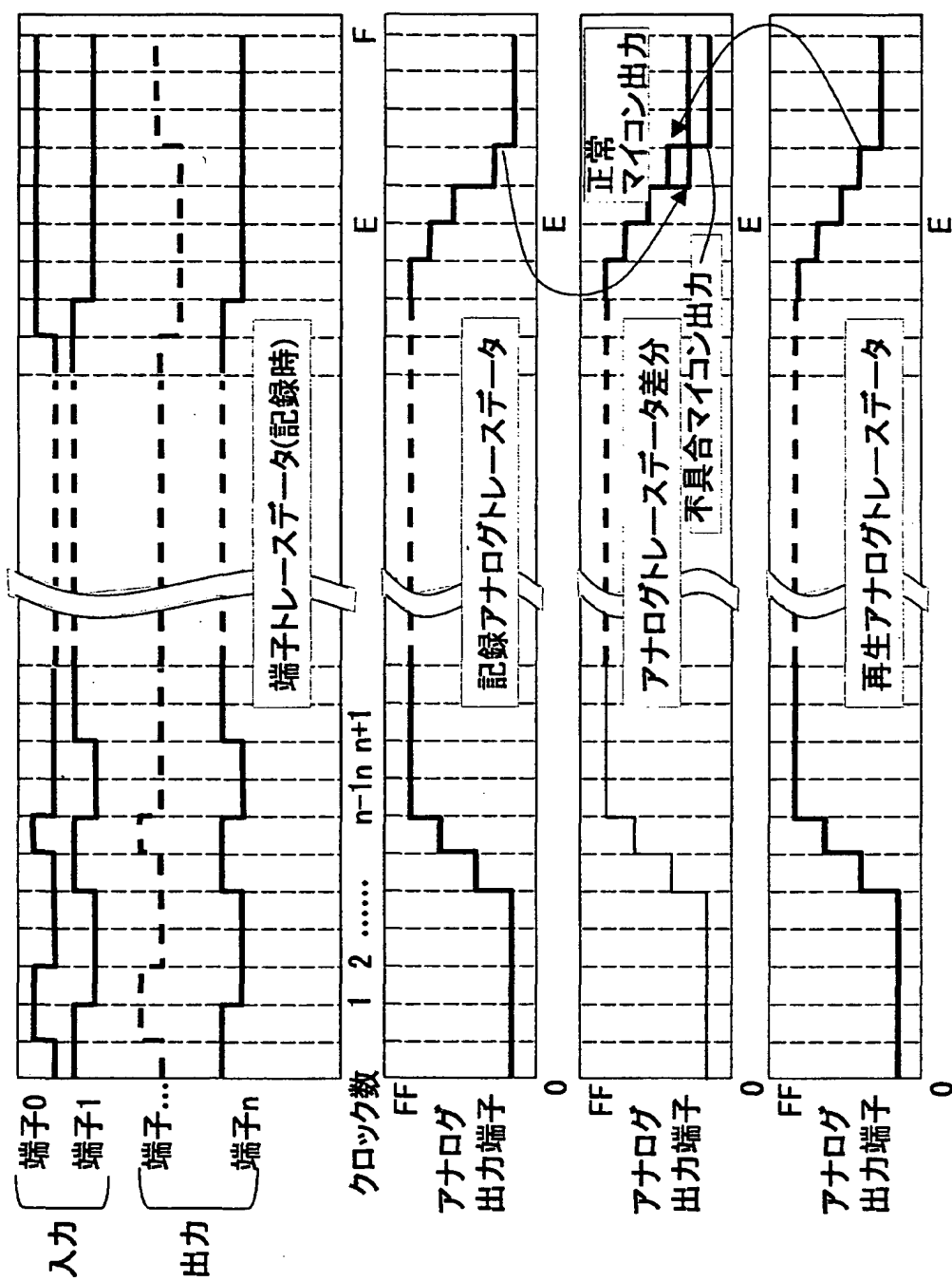
【図 40】



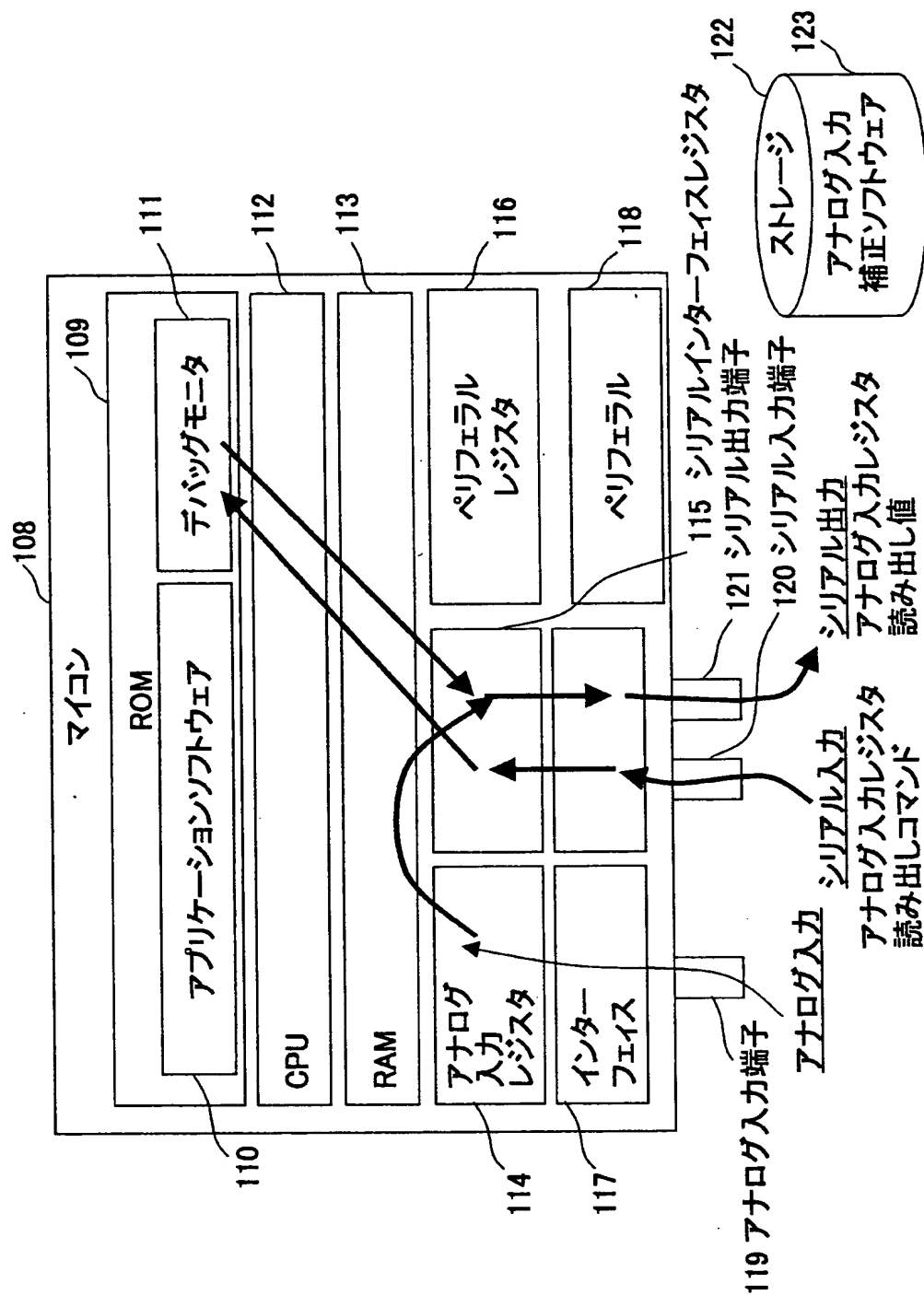
【図 41】



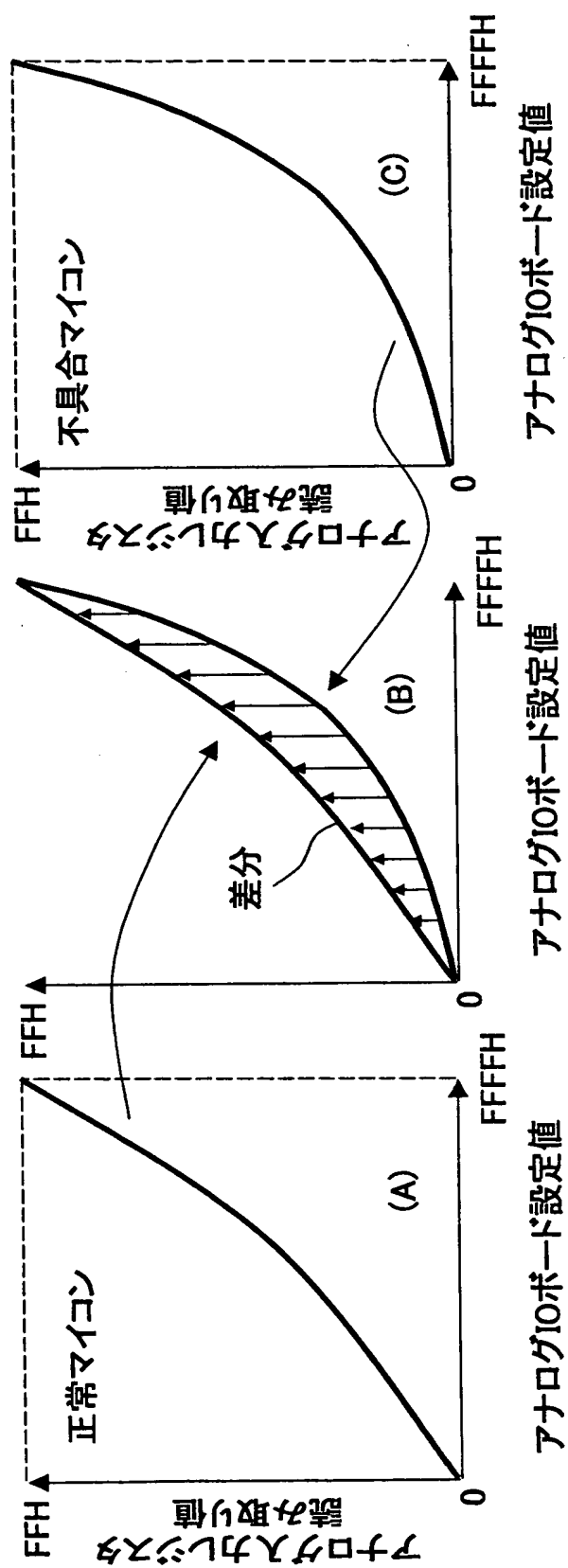
【図 4 2】



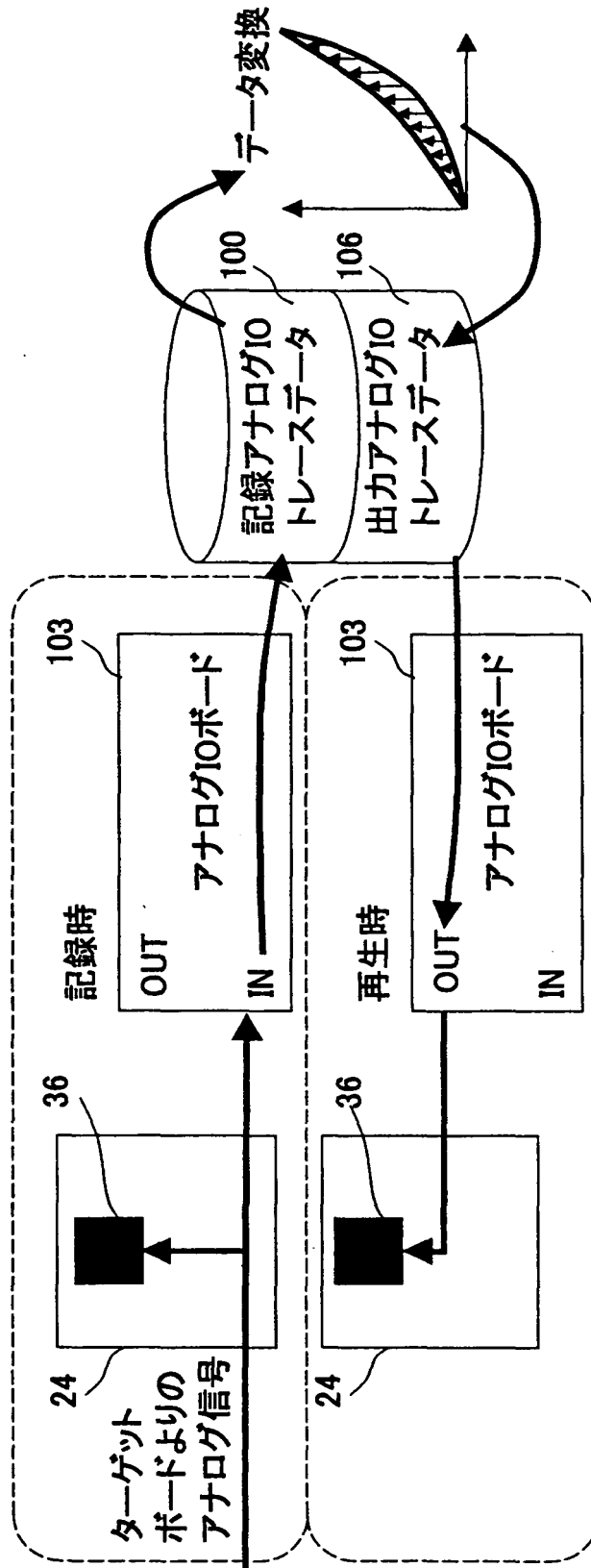
【図 4 3】



【図 4 4】

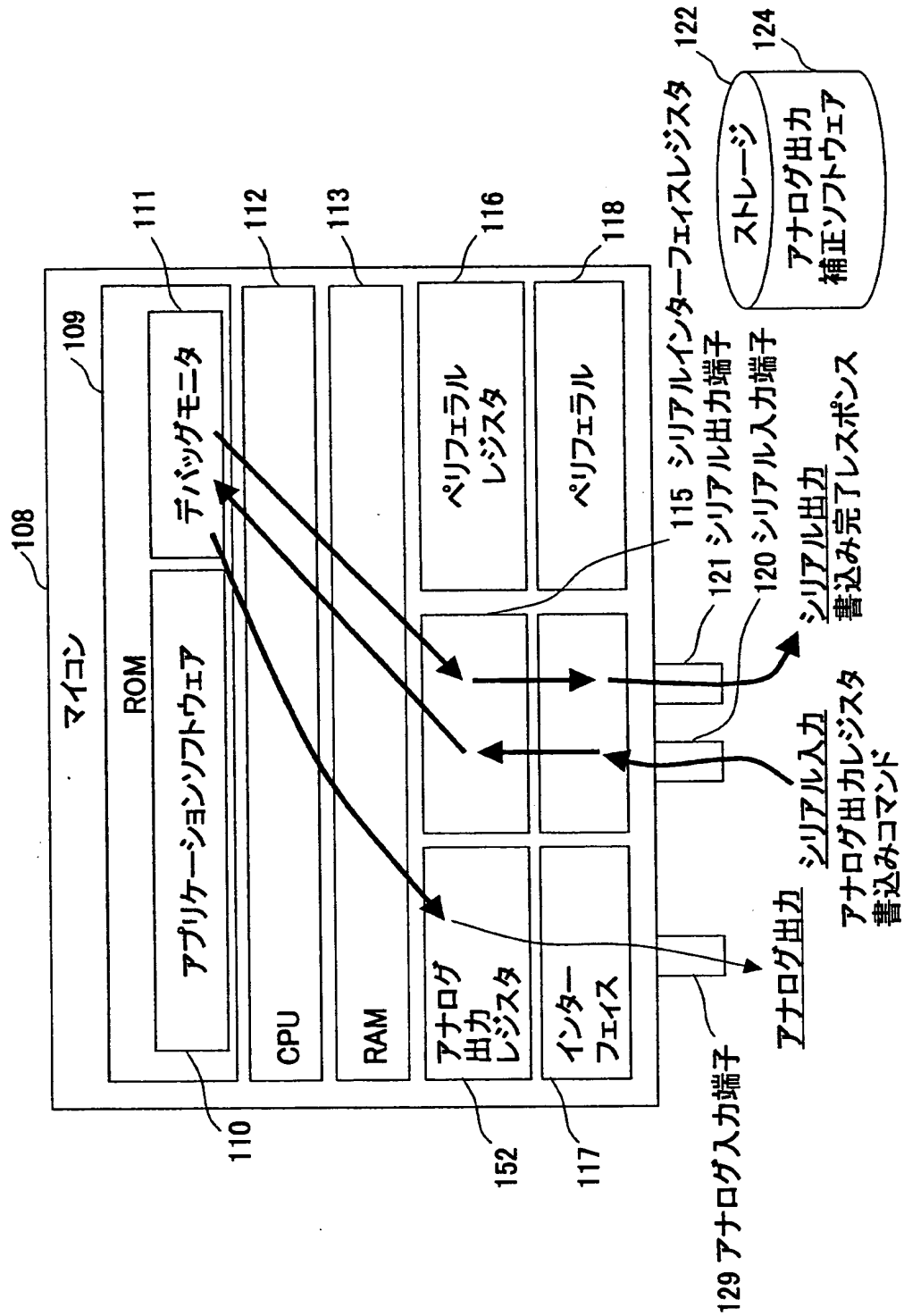


【図 45】

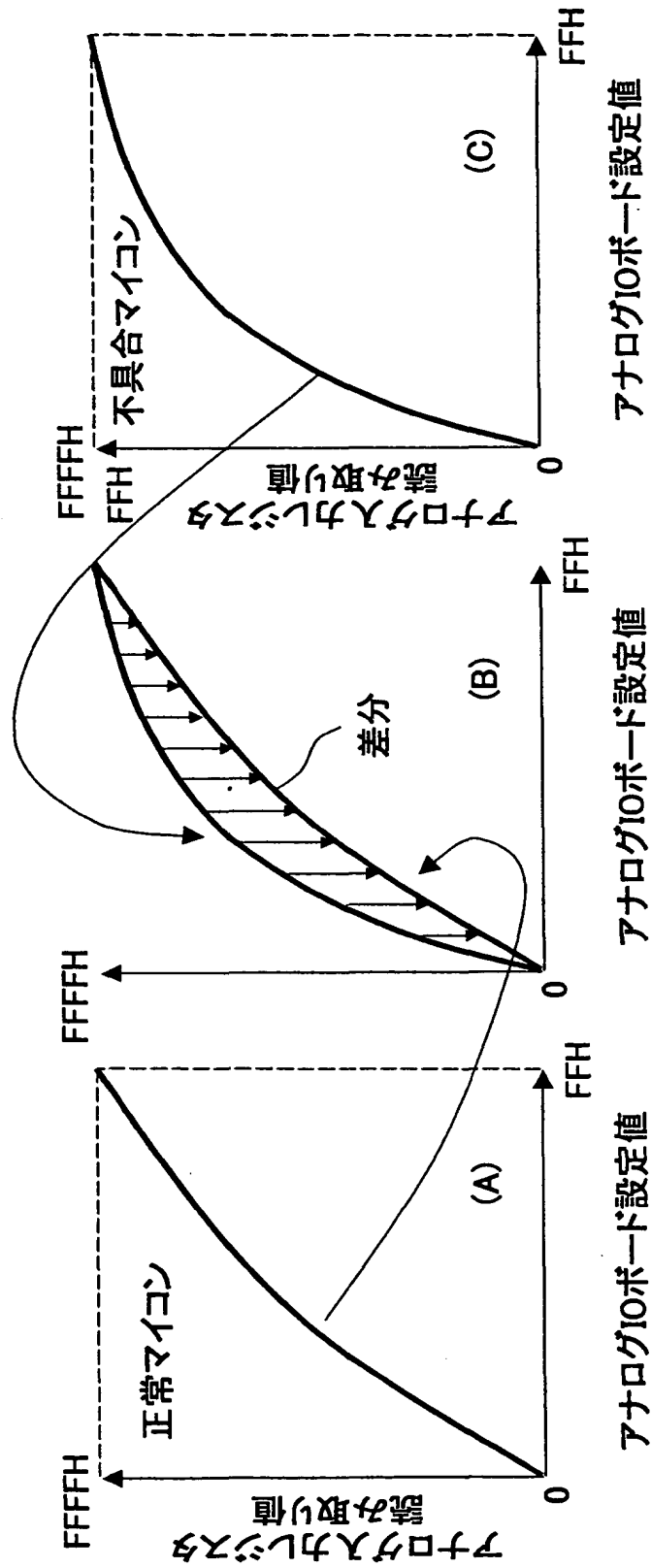




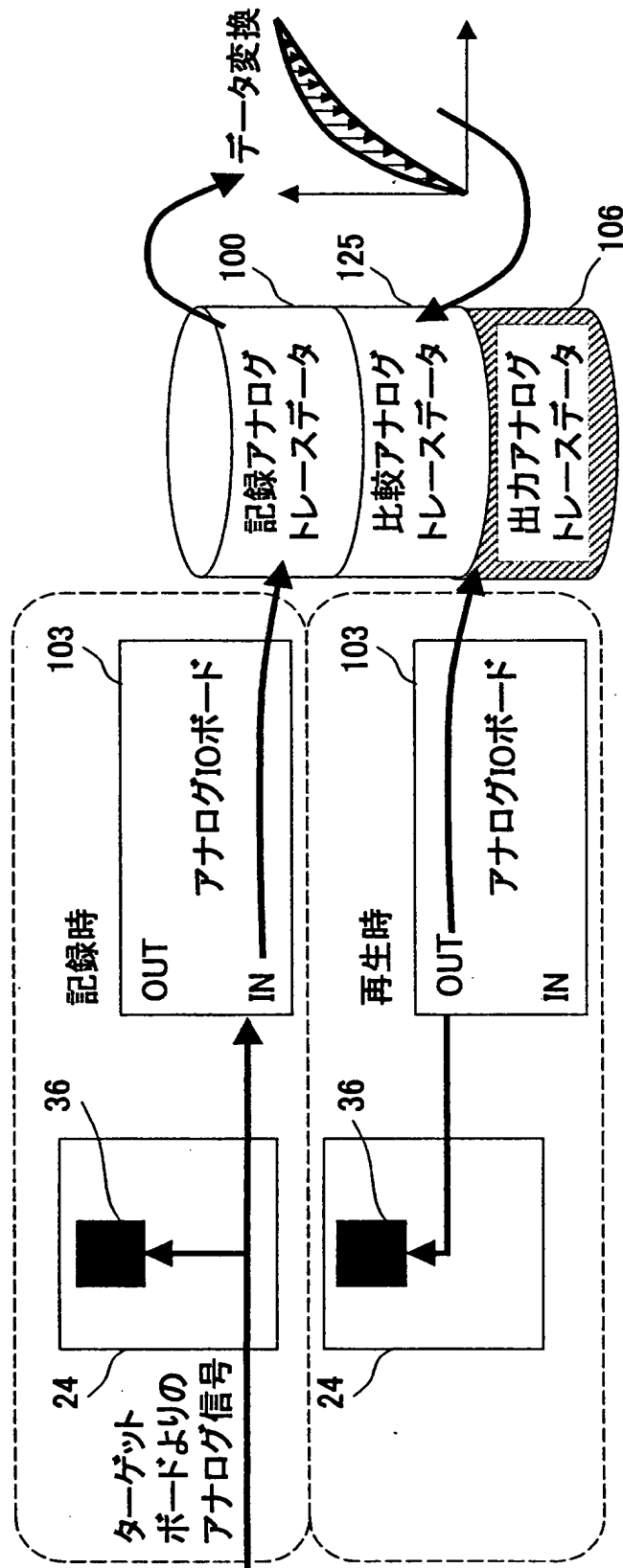
【図 4 6】



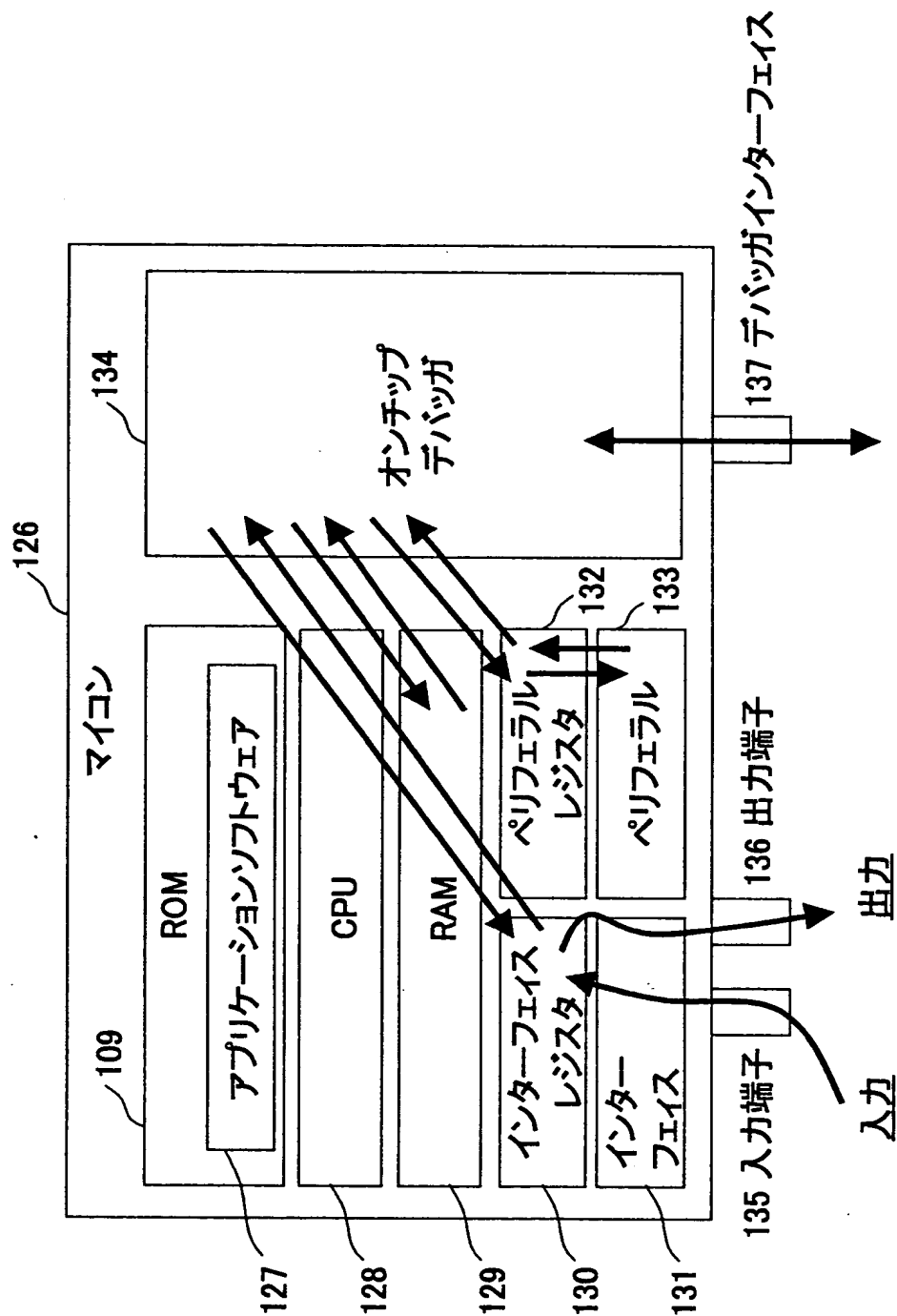
【図 4 7】



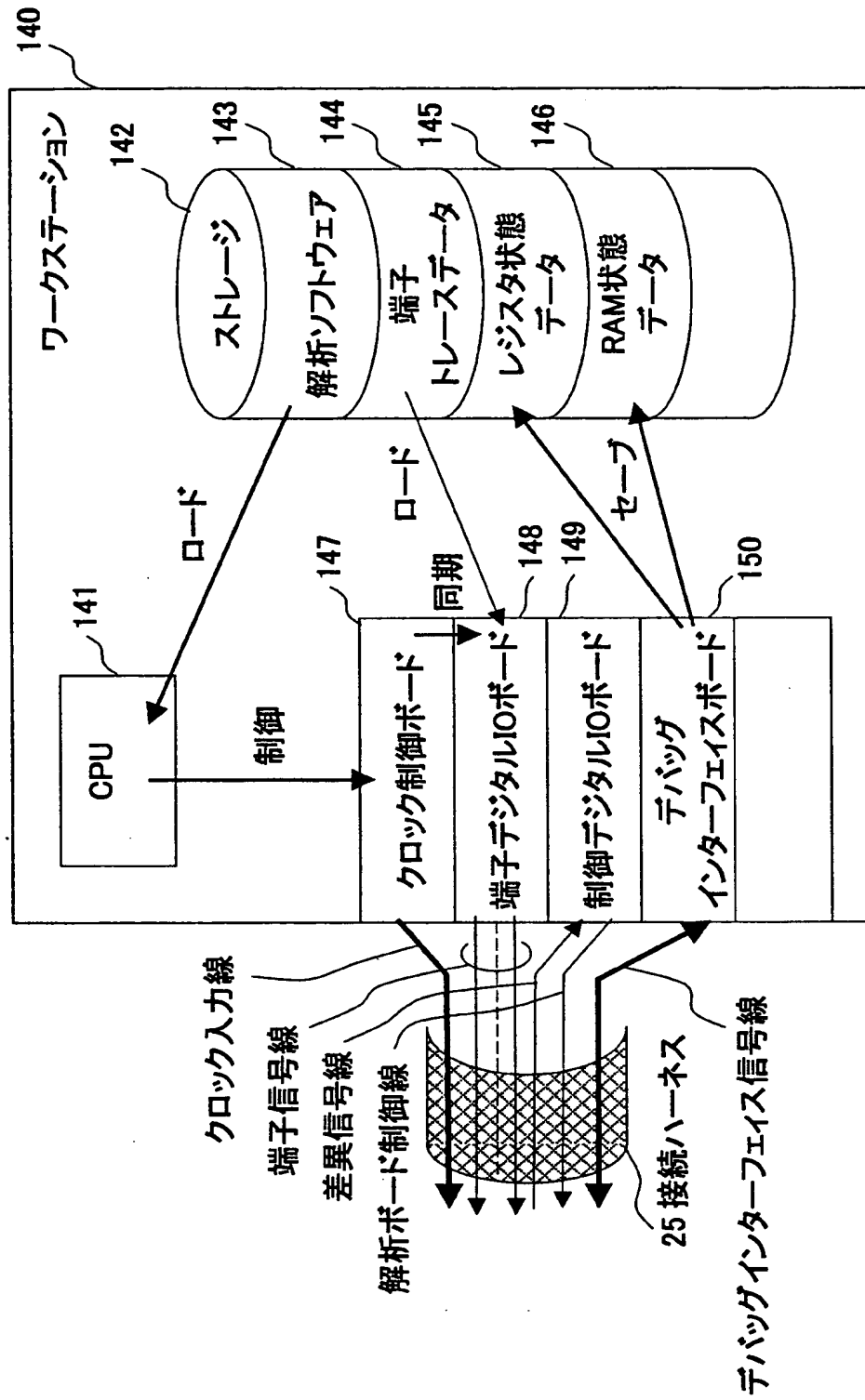
【図 48】



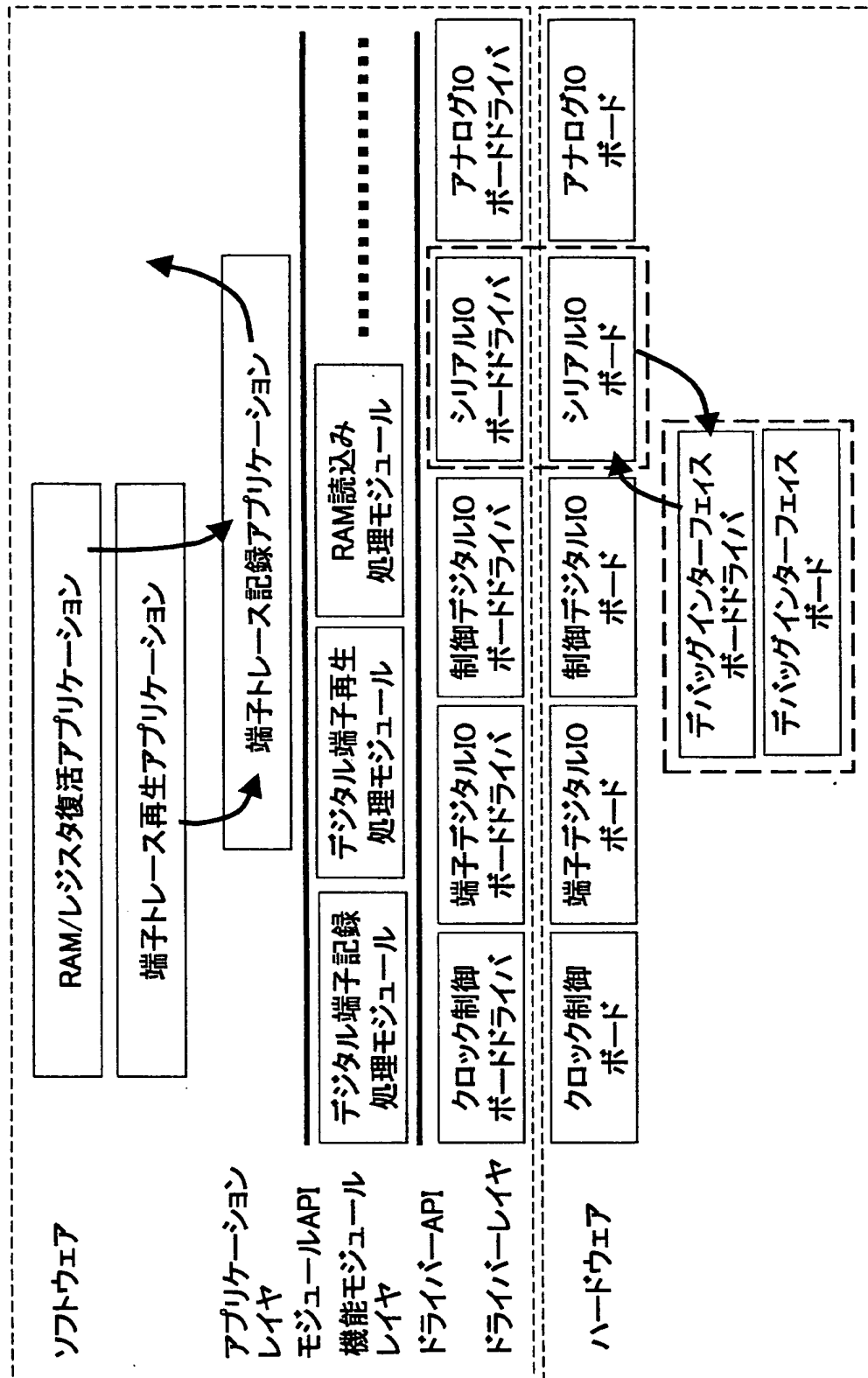
【図 49】



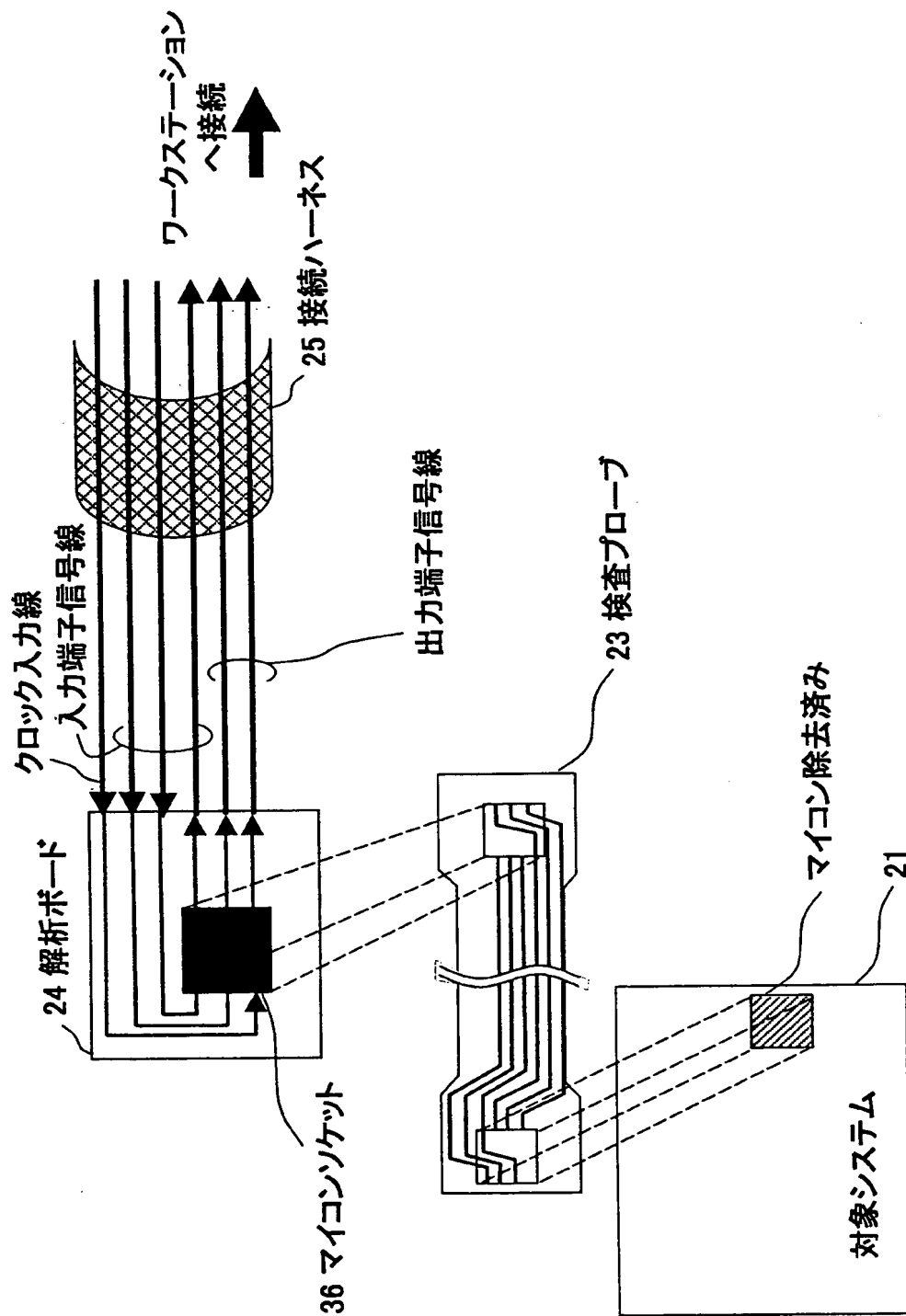
【図 50】



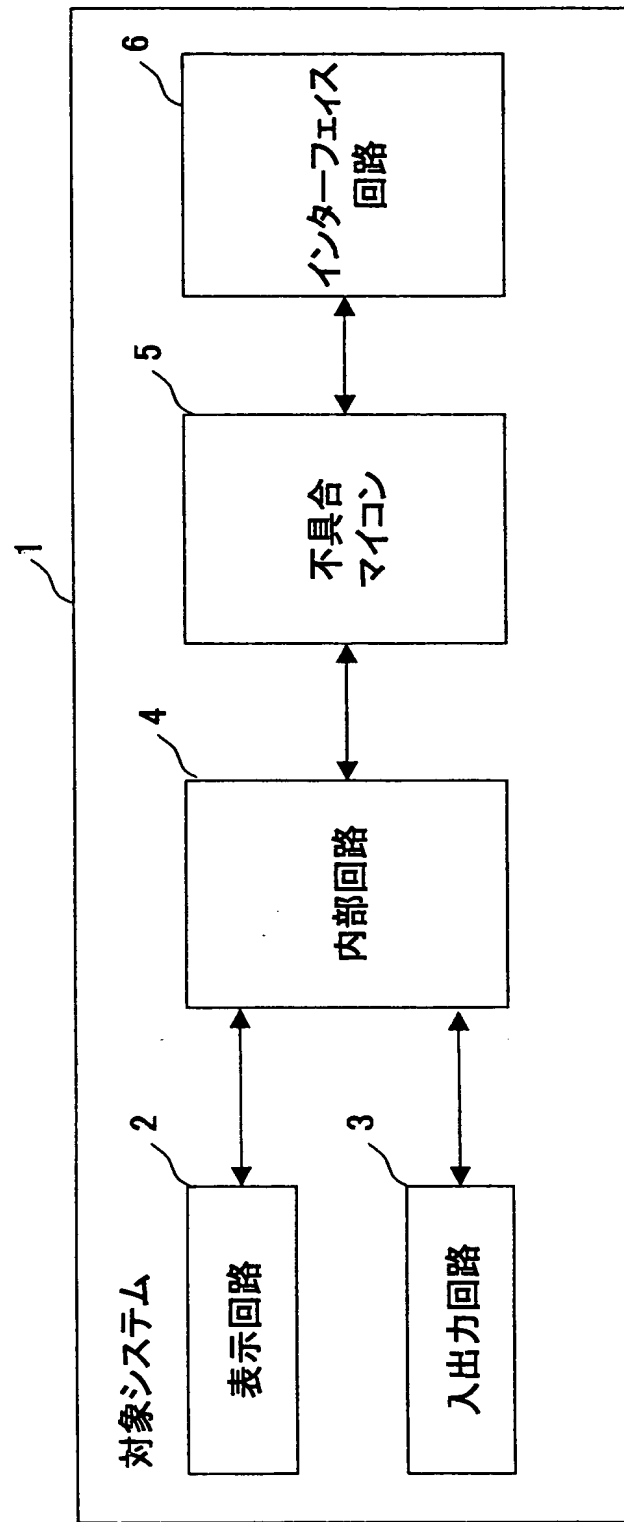
【図 51】



【図 52】

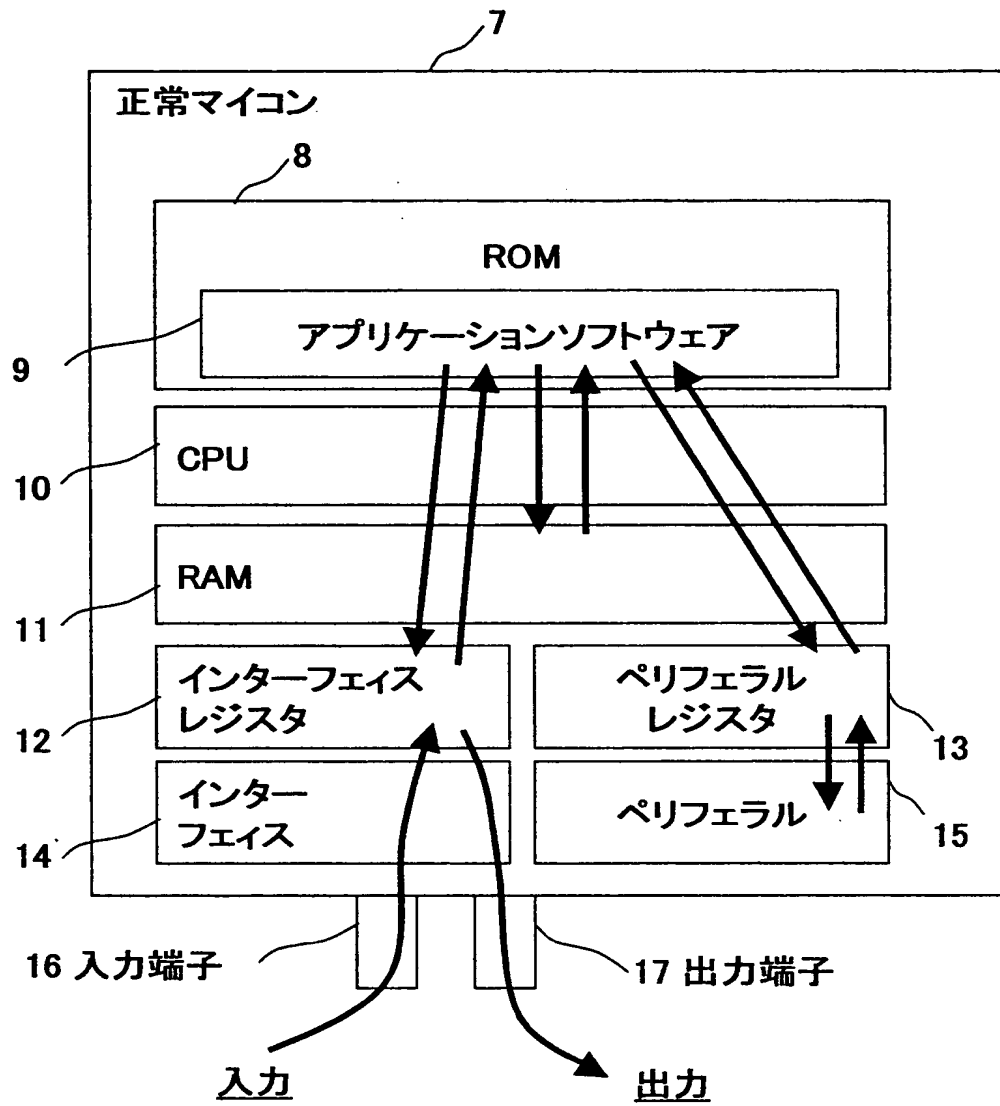


【図 53】

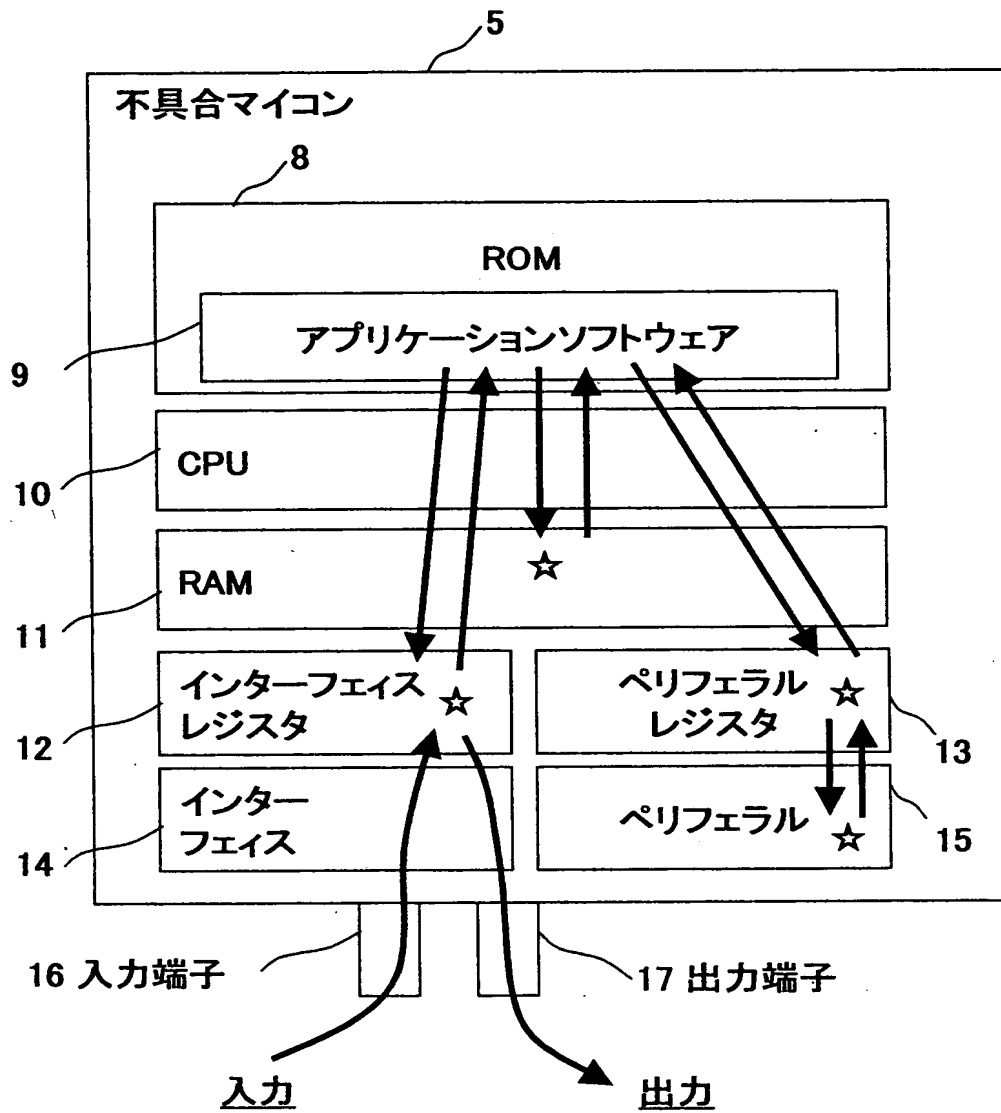




【図54】



【図55】



【書類名】 要約書

【要約】

【課題】 不具合の再現性が良く、真の不具合と検出された不具合との関連性の解析が容易である、ロジック L S I の不良解析システム、及び不良解析方法を提供する。

【解決手段】 ソフトウェアを内蔵したロジック L S I の不良解析システムであって、クロックに同期してロジック L S I の端子信号情報を記録する機能と、記録した端子信号情報をクロックに同期して再生する機能と、再生した端子信号情報と正常なロジック L S I の端子信号情報とを比較する機能と、解析対象のロジック L S I のトレースデータマップと正常なロジック L S I のトレースデータマップとのトレース差分マップを生成する機能を有することを特徴とする。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000000295]

1. 変更年月日 1990年 8月22日

[変更理由] 新規登録

住 所 東京都港区虎ノ門1丁目7番12号  
氏 名 沖電気工業株式会社

出 願 人 履 歴 情 報

識別番号 [ 5 9 1 0 4 9 8 9 3 ]

1. 変更年月日 1 9 9 9 年 6 月 1 7 日

[変更理由] 名称変更

住 所 宮崎県宮崎郡清武町大字木原 7 0 8 3 番地

氏 名 株式会社 沖マイクロデザイン